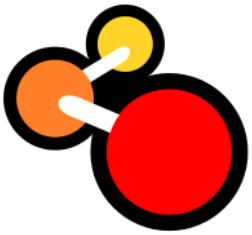


SecurActive

PV - User Guide Documentation

Release 3.3

by the PV Documentation Team

September 08, 2015

CONTENTS

1 Release notes **1**
- 1.1 What's new in 3.3 . . . 1
- 1.2 What's New in 3.2 . . . 1
- 1.3 What's New in 3.0 . . . 2
- 1.4 What's New in 2.18 . . . 2
- 1.5 What's New in 2.17 . . . 3
- 1.6 What's New in 2.16 . . . 3
- 1.7 What's New in 2.15 . . . 4
- 1.8 What's New in 2.14 . . . 4
- 1.9 What's New in 2.13 . . . 4
- 1.10 What's New in 2.12 . . . 5
- 1.11 What's New in 2.11 . . . 5
- 1.12 What's New in 2.10 . . . 5
- 1.13 What's new in 2.9 . . . 6
- 1.14 What's new in 2.8 . . . 6
- 1.15 What's New in 2.7 . . . 7
- 1.16 What's New in 2.6 . . . 7
- 1.17 What's New in 2.5 . . . 8

2 Use The PV Graphical Interface **11**
- 2.1 Access Through a Web browser . . . 11
- 2.2 Network Performance . . . 11
- 2.3 Application Performance . . . 12
- 2.4 Bandwidth . . . 13
- 2.5 Conversations, Flow Details & Raw Data . . . 14
- 2.6 HTTP Analysis . . . 14
- 2.7 SQL Analysis . . . 16
- 2.8 CIFS Analysis . . . 18
- 2.9 Matrix . . . 20
- 2.10 Top Reports . . . 22
- 2.11 DNS Performance . . . 22
- 2.12 TCP Events . . . 23
- 2.13 ICMP Errors . . . 23
- 2.14 Drill Down . . . 23
- 2.15 PDF / CSV Export . . . 25
- 2.16 Filters . . . 25

3 Main terms and concepts **27**
- 3.1 General Conventions . . . 27
- 3.2 Zones . . . 27
- 3.3 Application . . . 29
- 3.4 IP Merging . . . 30
- 3.5 Conversation . . . 30
- 3.6 Data Aggregation . . . 33

4 Metrics Computation **35**
4.1 Conversations . . . 35
4.2 HTTP . . . 36

5 Deployment **39**
5.1 How to integrate Performance Vision in your network? . . . 39
5.2 How to capture traffic? . . . 40
5.3 Supported Protocols . . . 41
5.4 Port-mirroring and duplicated packets . . . 43
5.5 Distributed Architecture . . . 45
5.6 Virtual Performance Vision . . . 48
5.7 Netflow . . . 50

6 Virtual Appliance Step-by-Step **53**
6.1 How to get the image of the Virtual Appliance . . . 53
6.2 Virtual Appliance Specifications . . . 53
6.3 Installation . . . 54
6.4 Validate the traffic capture . . . 61
6.5 How to use the product? . . . 62

7 Configuration **63**
7.1 Hardware . . . 63
7.2 Pulsar . . . 63
7.3 SPV Functional Configuration . . . 67

8 Interpreting the results **81**
8.1 Business Critical Application Dashboard . . . 81
8.2 Business Critical Networks Dashboard . . . 82
8.3 VoIP Module . . . 83
8.4 Application dashboards . . . 86
8.5 TCP Errors / Events . . . 91
8.6 Packet level analysis . . . 92
8.7 Interpretation Guidelines . . . 95

9 Licensing and Upgrades **113**
9.1 Performance Vision . . . 113
9.2 Deployment Mode . . . 113
9.3 Product Range Summary . . . 114
9.4 Hardware Versions . . . 115
9.5 VMWare Versions . . . 115
9.6 Performance Vision Versions . . . 116
9.7 How can I determine the model that is right for me? . . . 116
9.8 Limits . . . 117
9.9 License and Upgrade Installation . . . 117

10 Frequently Asked Questions **121**
10.1 Firefox freezes randomly on some pages . . . 121
10.2 Aggregate level changes when browsing from tables to charts . . . 121
10.3 How can SRT be greater than DTT ? . . . 121
10.4 How can we have 0 packets and no traffic at all on a conversation? . . . 122
10.5 What is this timeout column (in Analysis/TCP Error)? . . . 122
10.6 Why are some DNS request names missing? . . . 122
10.7 Some TCP conversations are reported twice, what's wrong? . . . 122
10.8 Pcap files generated by tcpdump are (mostly) empty . . . 122
10.9 How to do complex searches on domain names? . . . 122
10.10 How comes my VM keeps losing sync? . . . 123
10.11 What about Open Source? . . . 123
10.12 Standard TCP Session . . . 123

11 Known issues **125**
11.1 Configuration . . . 125
11.2 Interface . . . 125
11.3 Various . . . 125
11.4 Sniffer . . . 125
11.5 Upgrading . . . 126
11.6 Metrics . . . 126
11.7 Pulsar . . . 126

12 Glossary **127**

13 Appendix **131**
13.1 Integration with other Tools . . . 131
13.2 Custom Filters . . . 138
13.3 SPV For Developpers . . . 152
13.4 Protocol Stack List . . . 154
13.5 Licenses of open-source libraries . . . 155
13.6 CIFS Status Categories . . . 158

Index **177**

CHAPTER

ONE

RELEASE NOTES

1.1 What's new in 3.3

Full Operating System Upgrade

- The data disk has new partitions with a full new OS (Debian 8.0)
- Updated VMware tools
- Updated SSL libraries
- Fixed occasional Linux kernel bugs seen on some environments

Other Features & Improvements

- A new Top Database page in the SQL menu
- Support of MPLS
- Improved custom filters with now more than 500 research fields
- Pulsar Shell configuration now works with transactions
- Business Critical Applications (BCA) can be sorted by status (to get most degraded applications first)
- More flexibility with LDAP authentication
- Appliance monitoring data available in the configuration menu

1.2 What's New in 3.2

- **CIFS / SMB Performance Analysis (*CIFS Analysis*)**
  - Supports SMB 1.0, 2.0 and 3.0 (without encryption)
  - Decode file path, SRT, DTT, meta data commands, etc.
  - New Pages (CIFS): - Overview - Performance - Top IP Client - Top IP Server - Top Files - Top Trees - Top Users - Queries - Raw Data
  - CIFS custom filters (*Custom Filters*)
  - L3 - L7 links between Flows & Transactions
  - Switch from Flows to Transactions and switch from Transactions to Flows
  - Inline help and Protocol documentation
- **Other Features & Improvements**
  - LDAP Authentication
  - Second data merging level for http transactions for users with mirrored internet traffic.
  - User interface improvements, like the switcher widget for IP or Zones

  - Performance improvements for queries on large data range
  - Shellshock security update

1.3 What's New in 3.0

- Database Transactions Analysis
  - Supported Databases in Performance Vision for SQL Performance Analysis
    - Oracle
    - Microsoft SQL
    - MySQL (and derivatives)
    - PostgreSQL
- Multi-Node Analysis
  - Schedule packet captures on multiple nodes at once
  - Create triggered PCAPs at any time
- Links between Flows & Transactions
  - Switch from Flows to Transactions
  - Switch from Transactions to Flows
- New Features & Improvements
  - Top Protocol Stack
  - Top Source IP
  - Top Destination IP
  - Support For IEEE 802.1ah
  - Improvements in HTTP for IP origin address (servers & proxies)
  - Enrichment of existing views
  - Ability to cancel queries
  - Warning when a new version is available
  - New menu interface
  - User interface improvements
  - Database summary
  - Performance improvements on BCNs

1.4 What's New in 2.18

- Sniffer :
  - Support (beta) of the skinny protocol.
  - CSV dumper is now multi-core (better performances).
  - HTTP: capture all HTTP traffic. The page reconstruction can be activated by flagging zones.
- Applications :
  - New button to remove unused applications (and increase performance)

  - Help to create applications from NC flows
  - New default applications (more coherent, simpler and updated)
  - Applications *NC tcp* et *NC udp* have been removed
- Send an email if the license is invalid or the data disk is almost full.
- Advanced filters are multi-selectable
- New flow degradation configuration: ICMP. The configuration page has better inner documentation
- Kinetic Matrix (drag with the mouse to scroll throw matrices)
- New inline help in search forms for complex input like regex or custom filters
- New page in HTTP: Top URL
- *SPV For Developpers*

1.5 What's New in 2.17

- New applications (*Application configuration*):
  - Applications can be defined with much more criteria.
  - Applications can be exported to and imported from a CSV file.
  - A new configuration page allows to check application rules.
  - Webpattern and DynPort pages were no longer in use and were removed.
- New data field: Protocol Stack - which can be used to define an application.
- Deprecated URL pages were removed and replaced with new HTTP pages. Reports are automatically migrated.
- Non-IP flows are now integrated in all non-specific views from the *Application* and *Network* menus.
- Non-IP flows now have source and destination zones.
- New *Raw Data* pages to display data point to point chronologically, ie the way they are stored in the database without aggregation as much as possible.
- The sniffer now decodes HTTPS, a new page to set the SSL keys is available in the configuration (*TLS Decryption*).

1.6 What's New in 2.16

- Graphical interface:
  - Result columns can be retracted to give place for the other ones.
  - Normalize all search forms: use same filters on all pages when it makes sense.
  - Add a basic support of Netflow v5 ; add the filter "external capture" to filter on it.
- Huge performance improvement on CSV export of the database results.
- Final Custom filters: filters like *srt>500ms* are now available. (see *Filters*)

1.7 What's New in 2.15

- Transaction HTTP
  - HTTP transactions are activated for flagged `Zones` and `Applications` in the *Configuration* menu.
  - New chart showing the hits per status.
  - New *Hits* report page.
  - New *Top Host* and *Top Server* pages.
- A new filter input appears in most of forms: custom filters, see *Filters*.
- Config sniffer: more settings added.

1.8 What's New in 2.14

- [Sniffer] better sniffing and dumping performance.
- [Sniffer] more accurate SRT/DTT in presence of lost TCP segments.
- Transactions HTTP:
  - More thorough analysis of web applications.
  - New transaction querying mode, used in a new report page.
  - New chart: HTTP performance with *Page Load* and *Hit RT* over time.
- First step of a notification system:
  - Some events are now created by the different processes on the probe.
  - Those events are raised in a new information page in the *Configuration* menu.
  - Error events are reported in a status bar, available for the *administrator* group.
- Configuration of the data merging: over time, data lose precision. The loss levels are now configurable. You can choose to not merge at all.

1.9 What's New in 2.13

- [Pulsar] The command 'poller' allows to set the MTU of each poller.
- [GUI] Shows that an upgrade is incoming with the install logs, no more white screen.
- [Upgrade] Upgrade logs now have their own file: `/var/log/nova/install.log`.
- BCN workflow:
  - Added a performance chart displaying oriented RTT, DTT, RD and Retransmission rate.
  - New *Oriented Flows Details* page has more information about oriented conversations.
  - Advanced filters are now the same in the *Application* and in the *Network* menus.
- New Metrics displayed and filterable:
  - Diffserv: IP header to classify the flows, displayed in both *Flows Details* page.
  - MAC address: Ethernet addresses, displayed in both *Flows Details* pages.
  - OS: For TCP only, the new sniffer can detect the network fingerprint of a wide range of operating systems. Displayed in both *Flows Details* pages.

1.10 What's New in 2.12

- [Metric] Add metric MTU (Max Transfer Unit) in the Source/Dest Matrix and in Oriented Conversation.
- [Zone] New zone factory settings
- [License] New VMware versions: Free, NPS Express, NPS, APS Demo, APS Audit, APS Express, APS.
- [License] VMware APS exists in flows limit: Small, Medium, Large and Unlimited.
- [GUI] New chart that displays the 'Number of Flows'.
- [Data] Export data of graphics or tables as CSV files.
- [GUI] New advanced filters which all support "with" and "without" criteria.
- [GUI] New zone selector design in forms to show the zone tree.
- [Sniffer] Infinite loop of PCAP reading from 'pcap/replay/loop' directory.

1.11 What's New in 2.11

- [Matrix] Client / Server matrix, available metrics: - Traffic, - Packet Count, - Server Response Time (SRT), - Connection Time (CT), - Round Trip Time (RTT), - Data Transfer Time (DTT), - End User Response Time (EURT).
- [License] VMWare: per flow license model: - Limitation is set on central database sizing.
- [Reports] Added global and per page description.
- [GUI/Report] Logo is customisable and displayed in both HTML and PDF.
- [Report] Enhanced design.
- [GUI] IP Summary: Improved filtering. Charts are filtered according to the "zone" filter.
- [GUI] New matrix design.
- [GUI] Rework the menu to better suit the workflow.
- [Sniffer] Reduce memory consumption.
- [Sniffer] Huge disk IO reducing for *autopcap*.

1.12 What's New in 2.10

- [Matrix] Performance Mapping Breakthrough: new "Matrix" views. Select the metric of your choice and visualize at a glance where the issues are located.
- [Matrix] Source / Destination matrix, available metrics: - Traffic, - Packet Count, - Retransmission Rate (RR), - Round Trip Time (RTT).
- [GUI] One-click integration from Performance Vision to Wildpackets OmniPeek: take the best of both worlds. - High Level: Identify issues and drill down to specific traffic with Performance Vision, - Low Level: In-depth troubleshooting at packet level with Wildpackets OmniPeek.
- [Zones] High flexibility in network zone definition to easily deal with complex architectures. - Subnet - MAC Adress - VLAN - Listening Device - Poller
- [Zones] Support CSV-based zone definition through both import and export. Manage your network zones with the internal editor or use your own favorite tool (Excel or any other CSV-capable application).
- [Reports] Possibility to edit/modify parameters page per page.
- [Sniffer] Junkie is now in charge of zone/application tagging, so this work is distributed on each poller.

- [Sniffer] Support MTU metric.
- [Sniffer] Junkie now gets the PCAP files from *pcap/replay* directory as it was from a new network interface.

1.13 What's new in 2.9

1.13.1 New Features

- [Alerts] Business Critical Networks metrics are available through SNMP. The values can be queried through SNMP (Performance Vision MIB).
- [Metrics] Implementation of a new heuristic to find out clients from servers without 'SYN' packets.
- [Metrics] Support for HTTP chunked transfer encoding.

1.13.2 Changes

- [Reports] Queried time interval in reports has been simplified.
- [Reports] Email recipients have become optional as reports are now also stored on the probe and available through ftp.
- [Reports] Report edition now displays time intervals of each individual pages.
- [PCAP] The former limitation on storage size of manual PCAP files (20 GB) has been removed. User can now freely manage the size of captures depending on available storage capacity.
- [GUI] Time selection improvement.
- [GUI] In "Monitoring", information displayed by "top" screens has been harmonized.
- [Metrics] DTT will time out after 1 second with no data transfer. If no more data is received during this period, we considered that last packet received was the one to take into account for the DTT.

1.13.3 Major Bug Fixes

- [Metrics] Retransmission rate is now computed regardless of empty packets.
- [Metrics] The de-duplication process is no longer fooled by varying ethernet padding.
- [GUI] There were occasionally some empty lines in grouping tables.
- [Reports] Scheduling of report dates when set across two days (ex: from 23:00 to 01:00).
- [Reports] For reports, some client email applications were not displaying the PDF file attached.
- [PCAP] better *autopcap* performances when lots of files are generated.

1.14 What's new in 2.8

1.14.1 New Features

- [Alerts] Business Critical Applications metrics are available through SNMP. The values can be queried through SNMP (Performance Vision MIB).
- [GUI] Find the company vendor name behind a MAC address for non IP traffic.
- [Metrics] Added a new metric: "0-Window event" in TCP Events.
- [GUI] JavaScript performance improvements

1.14.2 Changes

- [PCAP] AutoPcap files are now kept for 72 hours instead of 48 hours.
- [Export] All data views can now be exported directly as a PDF page (new "Export as PDF" icon).
- [GUI] Updated TCP conversation workflow for an improved usability.

1.15 What's New in 2.7

1.15.1 New Features

- [Config.] POSIX regular expressions are available in web patterns.
- [Reports] Can now reorder pages in a report.
- [GUI] DNS resolution requests can now be done and undone with a button (column by column) and no longer through field mouse-over.

1.15.2 Changes

- [GUI] Replace in/out by srv/clt in all pages.
- [Metrics] Deduplication is now performed independently for every interfaces/VLANs if these are not aggregated.
- [Config.] Search and zone edition is now faster.

1.15.3 Major bug fixes

- [Metrics] SIP connections were not properly tracked in some cases.
- [Pulsar] Fix Pulsar `analyzer ifaces` and `help` commands.
- [GUI] Fix empty unfolded line bug in grouping tables.
- [System] Restart processes when they consume too much memory.

1.16 What's New in 2.6

1.16.1 New Features

- [GUI] User manual is now accessible from the GUI.
- [GUI] Advanced filters on client/server pages.
- [GUI] IP/subnet filter in "matrix" page.
- [GUI] Improved time frame selection with "last five used" history.
- [Pulsar] Pulsar now displays license information on the `poller` command.

1.16.2 Changes

- [GUI] "Top" screen reorganisation. We now have Tops for clients, servers, applications and ports.
- [GUI] ICMP messages regarding different connections are no longer merged.

1.16.3 Major bug fixes

- [Metrics] TCP keepalives no longer interrupt a data-flow.
- [Pulsar] Fix Pulsar `process` command.
- [GUI] Fix filters on unilateral flows or retransmission.
- [Reports] Fix missing columns in some reports.

1.17 What's New in 2.5

1.17.1 Installation notes

- Service Pack update must be installed before migrating from 2.x to 2.5. If the Service Pack is not installed, the 2.5 upgrade will not start.
- Migration must be done from a 2.x version. If you currently have 1.x version, please update first to version 2.0 or 2.3. Then, install the Service Pack, then install the 2.5 update.

1.17.2 New Features

- *Autopcap* for Business Critical Applications: available in Network conversation, DNS and VoIP depending on configuration. It works for both local and distributed environments.
- New Metric: DTT Client added to the several screens where the DTT Server was already present.
- New Protocols: LLMNR (Link Local Multicast Name Resolution), mDNS (Multicast DNS), NDNS (NetBIOS Name Service / WINS).
- Distributed poller management.

1.17.3 Changes

Network sniffing

- Automatically detects and listens again to network interfaces that come back up after a downtime period.
- At startup, automatically adjust and fine-tune deduplication parameters for the best balance between processing power required and deduplication efficiency.

Reporting

- User / Password / TLS security support.
- User can customize "From" field when sending a report.
- Reports stored as PDF files on the probe and available through ftp.

GUI

- For Business Critical Networks, the Retransmission Rate threshold can now be < 1%.
- Configuration area reorganized for clarity.
- In the Configuration area, delete buttons are now more intuitive.
- Animation when running a request (to avoid overloading the probe by launching the same request several times).

- The timeframe selection in the “Watch last” filter is now more intuitive.
- When a filter is set to some value, it will be highlighted to be more visible.
- In Non IP traffic screen, data can be filtered by MAC address.
- Bookmarked pages now have their own specific title instead of a generic name.
- In DNS screens, the filter on request types are now sorted alphabetically.
- New Screens
  - DNS Performance Graph, with DNS response times and number of packets over time.
  - TOP DNS Servers: DNS traffic and average response time sorted by server.
  - TOP DNS Clients: DNS traffic and average response time sorted by client.
  - DNS Overview
- New filters
  - Synthesis per DNS request types and DNS responses codes.

Pulsar

- “vpn” command has been renamed as “support”.

1.17.4 Major bug fixes

- Display of some charts could fail in some cases (long zone names added to long application names)
- Configuration was not correctly flushed in some cases.
- It was possible to define two applications on the same port for the same IP or subnet which led to approximate metrics for these applications.
- Oracle parser could stop working in some cases.
- Potential deadlock under intensive usage with the implication of several different parsers at once.
- Fix an issue with Flash player and Internet Explorer that forbids drill-down into graphics.

CHAPTER

TWO

USE THE PV GRAPHICAL INTERFACE

2.1 Access Through a Web browser

We assume here that the probe has been previously configured through the command line interface and the user knows the probe's IP address. The probe can be accessed either with `SSH` or with a Web browser. To connect with a Web browser, the ports to use are the `80`, `8080` or `443`. For physical, see *Configuration*. For virtual see *Virtual Appliance Step-by-Step*.

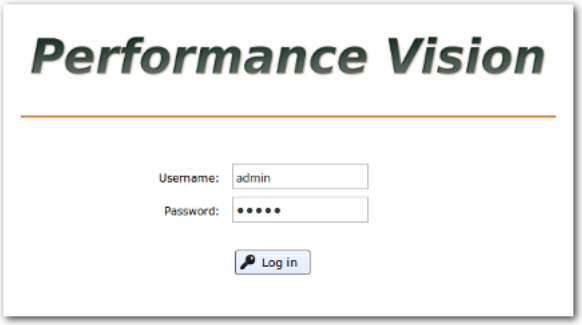


Figure 2.1: Login parameters in PV

If the IP address of the probe has been configured as `10.0.0.1`, then just open the URL `http://10.0.0.1` with your Web browser (or `https://10.0.0.1` to use the HTTPS protocol).

You can verify that you are actually connected to a Performance Vision appliance, by checking that the certificate serial number is `00:90:26:d5:46:2a:5e:66:ec`.

To log in, please use: `admin` as user and `admin` as password.

You are now logged in and ready to use the Graphical User Interface. For best performances, Mozilla Firefox is recommended.

2.2 Network Performance

Performance Vision provides a series of views to show how your network is behaving.

2.2.1 Business Critical Networks

Provided you have configured some critical networks (setting thresholds on volume and quality indicators between 2 zones), you will get a summary screen of the performance of your most critical network links on this screen. This is an auto-refresh screen, whose data can be integrated in your SNMP-based monitoring suite (if you enabled the SNMP daemon via *Pulsar*).

By hovering a specific time and link, you can view the origin of a degradation (latency, retransmission, excessive bandwidth consumption and in which direction it occurred).

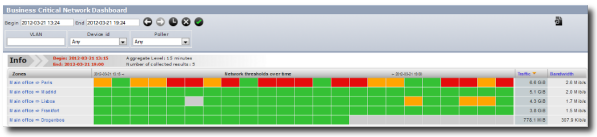


Figure 2.2: Business Critical Networks

You can access this view in the graphical interface in Dashboards / Critical Networks.

2.2.2 Performance over time chart

This view will show the main network performance metrics through time for a given selection (from one zone to another, for example): round trip time, retransmission delay, connection time, retransmission rate, volume of packets. This shows the evolution of the network performance; as in any view in Performance Vision, you can drill down to the conversation level by clicking through the graphs.

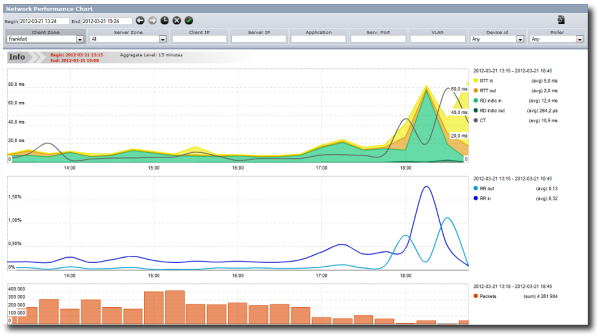


Figure 2.3: Performance over time chart

You can access this view in the graphical interface in Monitoring / Network Performance Chart.

2.3 Application Performance

Performance Vision provides a series of views to show how your applications are behaving.

2.3.1 Business Critical Application Dashboard

Provided you have configured some critical applications (setting thresholds on quality for a given application), you will get a summary screen of the performance of your most critical applications on this screen. This is an auto-refresh screen, whose data can be integrated in your SNMP-based monitoring suite.

By hovering a specific time and link, you can view the origin of a degradation (round trip time, server response time, data transfer time, quantity of transactions).

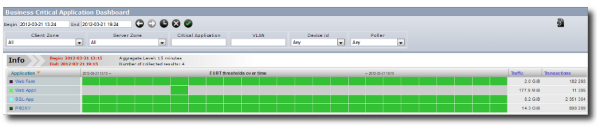


Figure 2.4: Business Critical Application

You can access this view in the graphical interface in Dashboards / Critical Applications.

2.3.2 Application Performance Dashboard

A simple click from the Business Critical Application Dashboard takes you to the Application Performance Dashboard. It shows you the evolution of the End User Response Time through time (along with the volume of transactions) and its breakdown in Round Trip Time, Server Response Time and Data Transfer Time. At a glance, you can understand the origin of a change in the End User response time.

Underneath this first graph, you find two additional bar charts, which help you understand which server(s) and Client Zone(s) are performing better / worse (and due to what component of the End User Response Time). The servers and zones are always presented from the one that corresponds to the highest volume of transactions to the lowest.

You can drill down and display either the Client Application Dashboard or the Server Application Dashboard by clicking on a specific server or client zone. This drives you to a specific application dashboard focusing on the same application for that specific server or client zone.

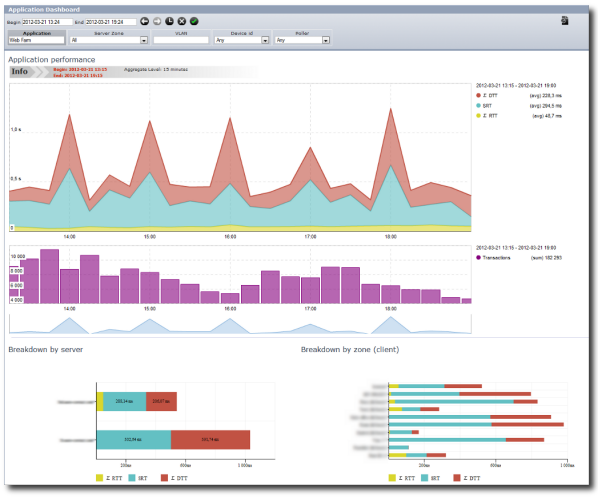


Figure 2.5: Application Performance Dashboard

This view is available for any TCP application in Dashboards / Application Dashboard.

2.3.3 Application Performance Chart

A more detailed view of the application performance is available here; it will show an even more complete set of metrics: RTT client & server, Server Response Time, Data Transfer Time client & server, retransmission rate, volume of packets.

Using filters, you can focus on a specific perimeter and view the evolution of the application performance through time. This view is specifically interesting to link the evolution of data transfer times to retransmission rates and data volumes.

This view is available for any TCP application in Monitoring / Application Performance Chart.

2.4 Bandwidth

You can graph the evolution of bandwidth through time.

From there, you can drill down to detailed conversations to display the main contributors of a peak of traffic, for example.

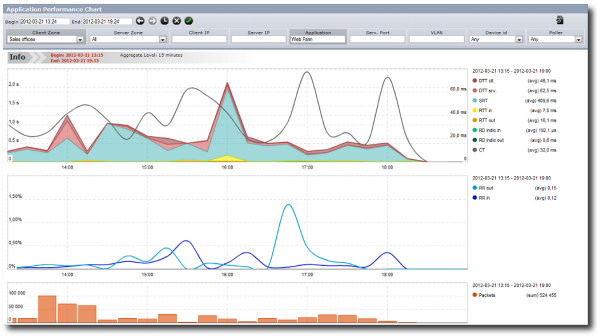


Figure 2.6: Application Performance Dashboard

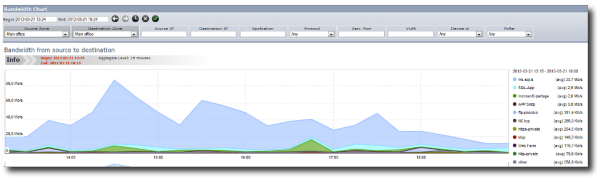


Figure 2.7: Bandwidth Graph

This view can be accessed through Monitoring / Bandwidth chart.

2.5 Conversations, Flow Details & Raw Data

A *conversation* represents the exchanges between two IP addresses. So why do we need *Flow Details* pages which are the same tables with more information?

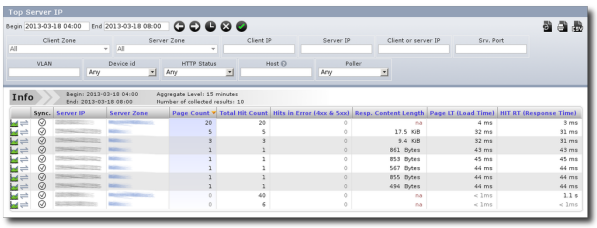
Even if both tables are alike, there is a subtle difference. By displaying more information (MAC addresses, detected OS, server port, protocol stack), the flow details may split a conversation in different rows and then the network and applicative metrics are split as well. *Conversations* is a more synthetic view whereas *Flow Details* is a troubleshooting one.

The Raw Data pages are different; the results show the data as stored in the database. The results are sorted chronologically by default. For instance, it is useful for troubleshooting to know in a global conversation where exactly the packets with a high metric value are.

2.6 HTTP Analysis

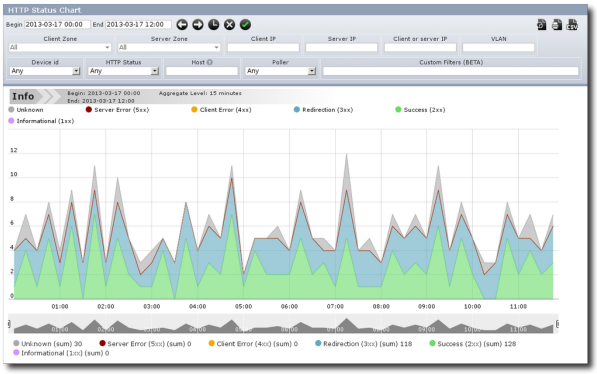
In the *Protocols* section, the set of pages for HTTP performance allows you to analyze HTTP traffic.

From these pages, you can easily find the most solicited servers or hosts, according to the number of hits, by payload or by response time.

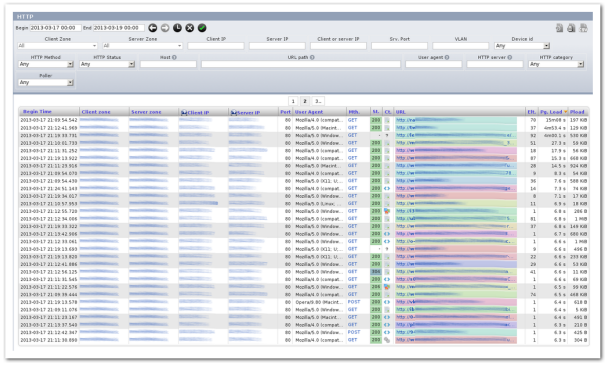


Using the graphs, you can check HTTP status and performance (including timing, error rate, payload size) over time.

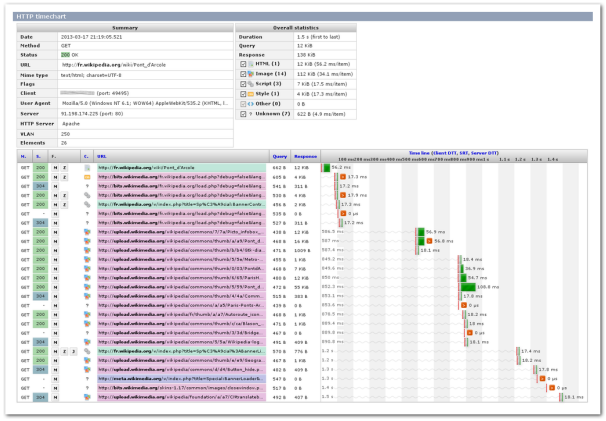
HTTP Performance Chart
Info
Page LT (Load Time)
HIT RT (Response Time)
Hits in Error (4xx & 5xx)
Page Count
Total Hit Count
Response Content Length Avg
Query Content Length Avg
Page LT (Load Time) (avg) 14.9 ms
HIT RT (Response Time) (avg) 407.3 ms
Hits in Error (4xx & 5xx) (sum) 0
Page Count (sum) 139
Total Hit Count (sum) 306
Response Content Length Avg (avg) 1.2 KiB
Query Content Length Avg (avg) 4.4 B


HTTP Status Chart
Info
Unknown
Server Error (5xx)
Client Error (4xx)
Redirection (3xx)
Success (2xx)
Informational (1xx)
Unknown (sum) 30
Server Error (5xx) (sum) 0
Client Error (4xx) (sum) 0
Redirection (3xx) (sum) 118
Success (2xx) (sum) 128
Informational (1xx) (sum) 0

You can view the captured HTTP transactions in detail by using the “Pages” and “Hits” pages. The first one lists all the HTML pages, while the second one gives you the details of every transaction (including image, javascript, css and other resources used to construct a page.)



When clicking a page, you get a timechart of all the transactions that occurred to build the page. From this view, you can inspect how the various servers involved were responding to the client’s browser. This allows you to get a visual overview of how the page was constructed over time.



You can also display the details of a single transaction by clicking on it. This will show a summary of the HTTP query and the response, in addition to the headers and an excerpt of the payload.

2.7 SQL Analysis

In the *Protocols* section, the set of pages for SQL performance allows you to analyze the SQL traffic. It shows you the queries with the usual metrics (responsiveness, payload size, SQL errors, etc) from the following database systems: SQL Server, PostgreSQL, Oracle Database or MySQL.

The poller should be able to decode the protocols from the following database systems:

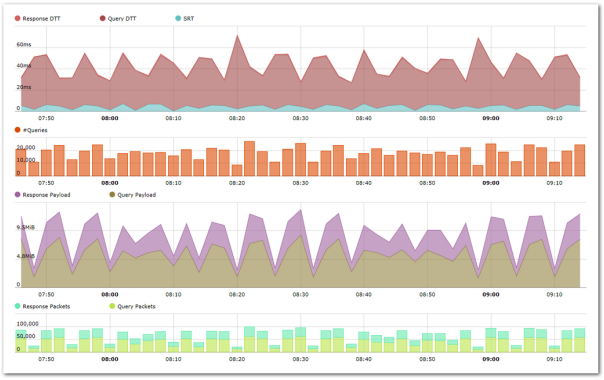
System	Support
SQL Server PostgreSQL MySQL Oracle	The poller should be able to decode the Tabular Data Stream (TDS) protocol from version 7.0 to 7.4 which corresponds to SQL Server 2000 to SQL Server 2012. Only the protocol 3.0 is supported which is implemented in PostgreSQL 7.4 and later. The protocol v10 is supported which is implemented by MySQL 3.21.0 and later. Oracle database uses the Transparent Network Substrate protocol (TNS). Since this protocol is proprietary and almost no technical specification is available, the decoding is best effort. It has mainly been tested on Oracle 10g and 11g with JDBC drivers.

By default, for performance reasons, and to keep the storage to a reasonable size, only the type of query (e.g. SELECT, UPDATE,..) is kept in aggregate levels above 2 minutes. You can configure this behavior in the "Data Merging" configuration page.

2.7.1 Graph

The main SQL graph allows you to view the performance of your SQL queries over time.

You can graph the performance of a specific query or a specific server using the filters.



2.7.2 Top Views

You can request the *top* most solicited server (according to the number of queries or total payload, for example).

You can request the list of queries that occur most often, or which take the longest time to get a response.

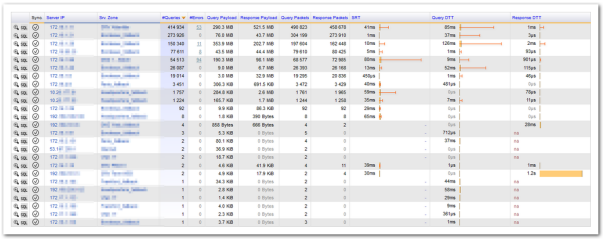


Figure 2.8: The *top* servers (most solicited) per number of queries.

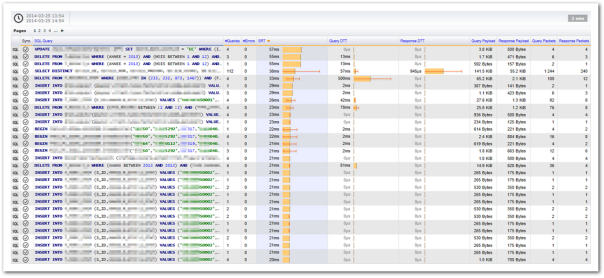


Figure 2.9: The queries that have the highest SRT measure.

2.7.3 Queries

You can also browse the queries over time.

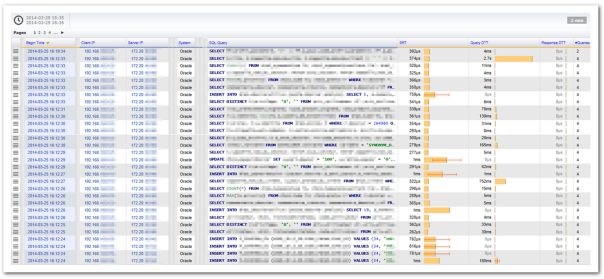


Figure 2.10: The queries over time.

2.8 CIFS Analysis

In the *Protocols* section, the set of CIFS performance pages allows you to analyse the CIFS traffic. CIFS includes SMB v1 to v3 protocols. It shows the CIFS commands with the usual metrics (responsiveness, payload size, ...) and some specific ones like metadata payload or data size effectively written by the server. Of course, when a file is handled, its path and tree will be available.

The CIFS set of protocols contains plenty of commands and statuses. In Performance Vision, we have classified the statuses in three categories: 'success', 'warning' and 'error'. You can find the details of how each status is classified in the appendix *CIFS Status Categories*.

We defined a category of common statuses, containing the most common CIFS errors and warnings. The list contains the following statuses:

- STATUS_NO_SUCH_FILE,
- STATUS_NO_SUCH_DEVICE,
- STATUS_OBJECT_NAME_NOT_FOUND,

- STATUS_OBJECT_PATH_INVALID,
- STATUS_OBJECT_PATH_NOT_FOUND,
- STATUS_OBJECT_PATH_SYNTAX_BAD,
- STATUS_DFS_EXIT_PATH_FOUND,
- STATUS_REDIRECTOR_NOT_STARTED,
- STATUS_TOO_MANY_OPENED_FILES,
- STATUS_ACCESS_DENIED,
- STATUS_PORT_CONNECTION_REFUSED,
- STATUS_FILE_DELETED,
- STATUS_INSUFF_SERVER_RESOURCES,
- STATUS_MORE_PROCESSING_REQUIRED,
- STATUS_BUFFER_OVERFLOW,
- STATUS_WRONG_PASSWORD,
- STATUS_NETWORK_ACCESS_DENIED,
- STATUS_TOO_MANY_SESSIONS.

To filter results by this category, use the following custom filter:

```
cifs.status = "common"
```

The drilldown workflow of the CIFS metric starts with an overview of the different commands and a chart of performances over time, then continues with the Top pages: *Top IP*, *Top File*, *Top Tree*, *Top User*. The troubleshooting pages are *Queries* and *Raw Data*.

Note: In the *Top Files* page, the CIFS queries without any file path are removed from the results.

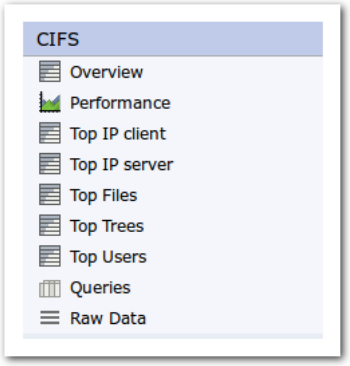


Figure 2.11: The CIFS Menu, from Overview to Raw Data.

2.8.1 Graph

The performance graph shows lots of metrics over time. You can compare applicative performance such as DTT and SRT with the number of queries, the payloads in each direction and finally the applicative packets.

2.8.2 Queries

The *Queries* page shows all CIFS transactions in detail. It can display the CIFS transaction information like *User*, *Domain*, *File Path*, *Command*, *Status* with their relative performance metrics *SRT*, *DTT* and their associated deviation.

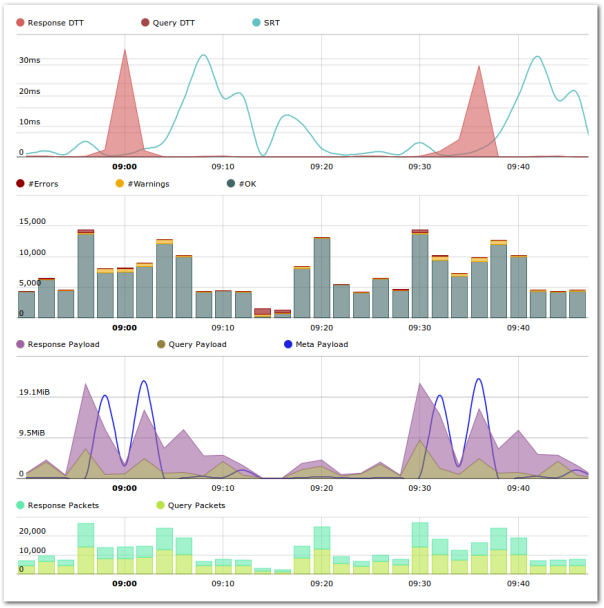


Figure 2.12: The CIFS graph with its four charts.

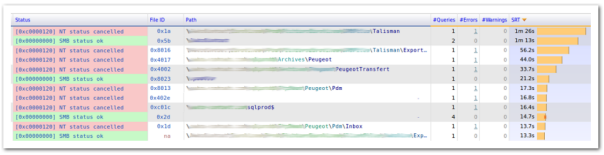


Figure 2.13: CIFS queries showing Status, File Path and SRT.

2.9 Matrix

The matrix view provides a representation of various metrics, where every cell represents the value of this metric from one zone to another zone. Each cell can contains extra values to better interpret the result (such as the number of packets used to compute a mean, and so on). This report provides a very synthetic view of the mapping of the metric which is observed.

The matrix can be used both for Client/Server and Source/Dest observations. The Client/Server matrix can be found in the *APPLICATIONS* section, while the Source/Dest matrix can be found in the *NETWORK* section.

The matrix is presented as follows:

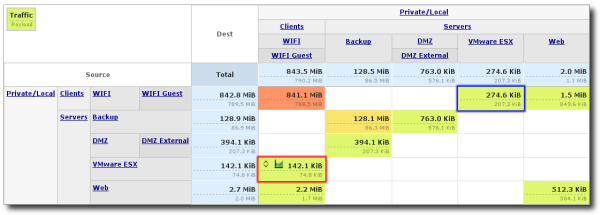


Figure 2.14: The *detailed* Matrix

The matrix will show a mapping of all flows as follows:

- *blue* cells represent the total for a zone (the sum of all the values in a row or in a column).
- *green* to *red* colored cells represent the traffic from one zone to another zone. The color represents the relative value regarding the maximum value of the whole matrix (*red* is the largest value displayed).

In the matrix above, we can see that there was 142KiB of traffic from machines in the `ESX` zone to the machines in zone `Wifi Guest`. The opposite direction, shown in blue, tells us that the traffic on the machines from the `Wifi Guest` zone to those in the `ESX` zone amounted for 274KiB.

There are two types of matrix presentations:

- the *detailed* matrix, which displays a zone and all its child zones, show how zone is spread in its subzones.

  The matrix above is such a matrix, where the zone *local* was chosen both for source and destination. We can see the zone and all its child zones.

- the *contextual* matrix, which displays a zone among its ancestor zone. This is convenient to check which part of the network is related to a specific zone.

An example of a *contextual* matrix that allows us to check how the *local* zone fits into the whole configuration:

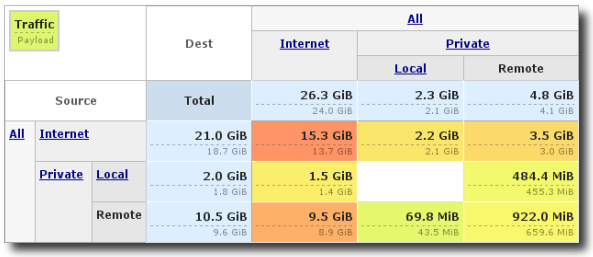


Figure 2.15: The *contextual* Matrix

The navigation within these matrices are thus different. The *detailed* matrix allows to select the zone to display (ignoring all other ones), while the *contextual* zone allows to select which zone to focus among its ancestor zone.

You can filter the flows taken into account by defining:

- the *observation period*
- the source *zone*
- the destination zone
- the *application*
- and other common filter such as *VLAN*, *poller* and so on.

Another matrix example:

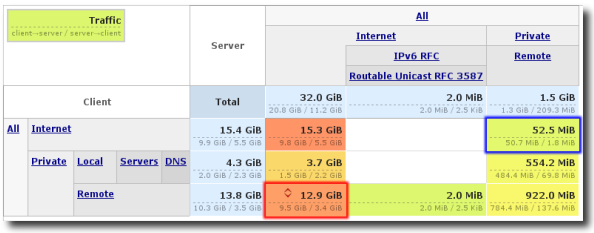


Figure 2.16: The *detailed* Matrix in the Client/Server context

In this matrix, we are presented with a different view of the metrics. Here, we can observe in the red cell that all of the communications initiated from machines in the `Remote` zone to the machines in zone `Internet` accounted for 12.9GiB (**total for both direction**). Meanwhile, in the blue cell, the communications initiated by machines from zone `Internet` to those in the zone `Remote` accounted for 52.5MiB.

2.10 Top Reports

You can easily get the top clients, servers, applications for any traffic (all or a specific application, zone, etc.)

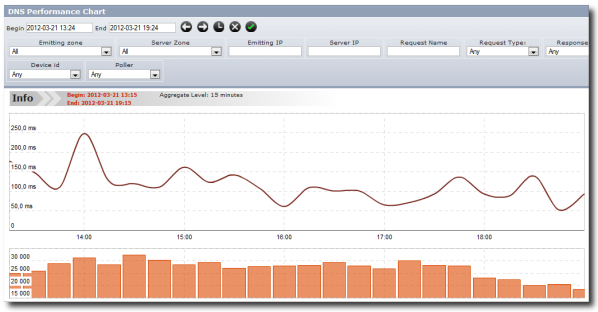
You can sort each top on the most adequate criteria (volume, sessions, SYNs, etc.)

This view can be accessed through Monitoring / Top Reports.

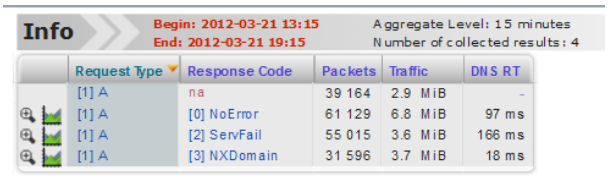
2.11 DNS Performance

Performance Vision provides an in-depth view of name resolution events and performance (for DNS, Netbios, mDNS, etc). When troubleshooting, this view can display:

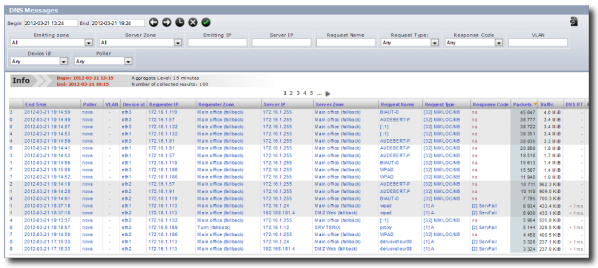
- The evolution of the DNS activity (an excessive peak may reveal a misconfiguration / infection)



- The evolution of DNS response times which impacts the quality of experience of end users.
- Unexpected name resolution protocols (are you still using Netbios/WINS, when you thought you only rely on pure DNS? Do you have more DNS requests in error than successful ones?)



- Are some of my hosts trying to resolve out of abnormal servers? (Rest of migrations, misconfigurations, infections)
- Can I see hosts with abnormal request volumes? (infection, misconfiguration)



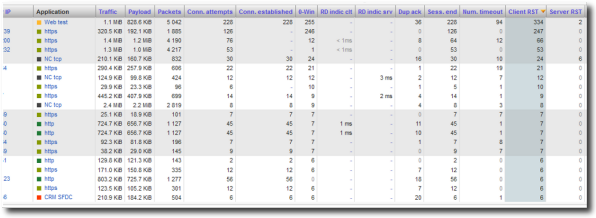
- Have I got some configuration issues (short TTL values, lack of caching)? Look at the DNS conversations with the largest number of transactions.

This view can be accessed through Diagnostic / DNS.

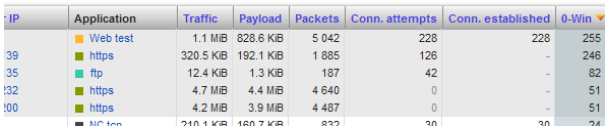
2.12 TCP Events

Performance Vision provides an in-depth view of TCP anomalies and events. When troubleshooting, this view can display:

- TCP conversations where the sessions are not ended correctly (Timeouts, RSTs, etc) This may help you understand when you can observe disconnections, if the client or server side is responsible for it.



- Bad transmission rate: if the data transfer is slow for a specific application, it may, of course, be due to network congestion, retransmission issues, but also to TCP errors like 0-Windows. By looking at specific conversations, you can view whether the TCP window is being reduced and by whom (client / server).



IP	Application	Traffic	Payload	Packets	Conn. attempts	Conn. established	0-Win
	Web test	1.1 MiB	828.6 KiB	5 042	228	228	255
39	https	320.5 KiB	192.1 KiB	1 885	126	-	246
35	ftp	12.4 KiB	1.3 KiB	187	42	-	82
32	https	4.7 MiB	4.4 MiB	4 640	0	-	51
00	https	4.2 MiB	3.9 MiB	4 487	0	-	51

- Abnormal behaviors: by sorting the TCP events by number of SYN packets, you can easily view which machines are generating a very high volume of TCP session start, which eventually do not drive to a complete TCP session setup. If you see machines with large volume of SYN packets and few / no session setup, these machines are either misconfigured or infected.

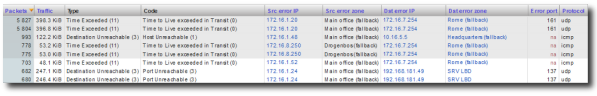
This view can be accessed through Diagnostic / TCP events.

2.13 ICMP Errors

Performance Vision provides an in-depth view of ICMP errors. ICMP errors will report the volume of flows which cannot be set up (either because the network, host, or port is unreachable). This can reveal:

- An unavailable host
- A network which is not reachable (either it does not exist - which reveals a configuration / infection issue on the source host, or it is not available - configuration issue?)
- A port which is not reachable (either the source machine is scanning or it is misconfigured and tries to reach a service which no longer exists or has been migrated).

This view is great to pinpoint configuration and infection issues.



This view can be accessed through Diagnostic / TCP events.

2.14 Drill Down

You can navigate between the metrics to drill down to the point where you want to analyze fine details about performance problems, or the opposite, to explore the context of such problems. You can also easily navigate to a different related metric.

You can see several labeled icons at the beginning of most result tables, that direct you to a more detailed / more general page matching a particular line.

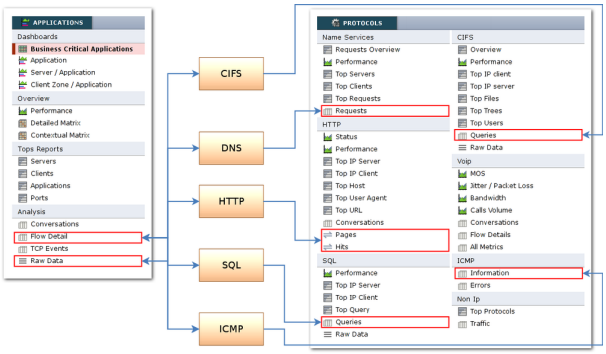


Figure 2.17: Overview of the relation between the metrics.

If some TCP connections are slow, it is also possible to go directly to the protocol analysis if available. If they're actually SQL connections, an SQL icon will be available on the corresponding lines, and clicking it you will see all the SQL analysis that matches these TCP connections. Likewise for HTTP transactions or other metrics, we are able to analyze in details.

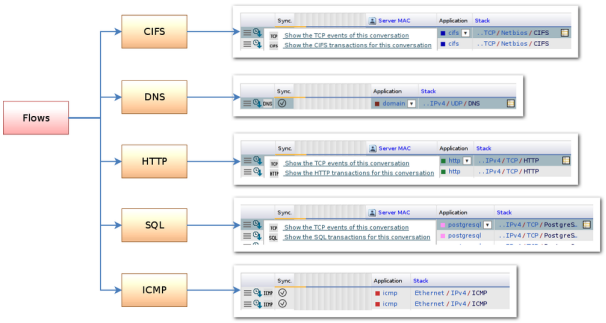


Figure 2.18: Links from flow metrics to detailed metrics.

Note: Drilldown can return no data for several reasons:

- The selected transaction does not match an activated *Zone* (for metrics like SQL or CIFS).
- No response has been parsed for the transaction, check the client and server packet counts for unilateral flow.
- The payload does not generate a metric flow, like keep-alive or notifications for CIFS.

A navigation between metrics that are related is available. For example, you can obtain the DNS queries that are related to a HTTP connection by clicking the "DNS" link in one of the result in the "Pages" view of the HTTP protocol section.

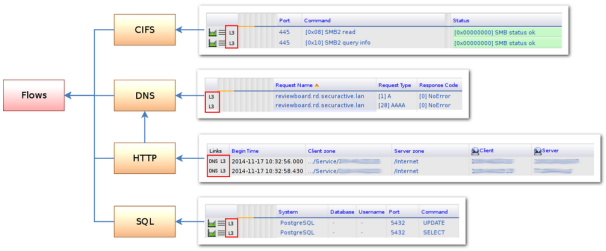


Figure 2.19: Links from detailed metrics back to flows.

2.15 PDF / CSV Export

On any web page of the web interface (except for the *Configuration* section), you will have the ability to export data either in PDF or CSV formats. As long as no query has been performed, the export buttons remain deactivated. Once a query has been run, the buttons are activated and you can export the data in the format you prefer.

In PDF, you will have a PDF document presenting the same data (tables or graphics) as the ones you got on your browser page.

In CSV, you get a text file with the corresponding data values. You can then use this CSV file on any of your own data processing. In case you have several graphics or tables on a page, you will have to choose the one you want to export the associated values.

Note: Please note that you cannot export the Matrix views into CSV files; the hierarchical nature of the Matrix does not fit into the flat structure of the CSV format.

2.16 Filters

For the full technical documentation, please see the appendix *Custom Filters*.

In each report page, you can filter the query on different fields. All filters will be combined with the `AND` operator. When you set a filter and send a request, then this filter is saved for the current session.

For a more complex search, you can use the *custom filters* input. In this field, you can combine filters with any logical operators: (`OR`, `AND`, `NOT`) and can order subexpressions using parentheses. You can filter on most of the common available fields.

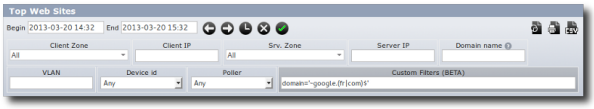


Figure 2.20: Custom Filters Example

Below are some of the available fields (the full list is in *Custom Filters*)

- app,
- capture.begin, capture.end,
- device,
- diffserv, diffserv.clt, diffserv.srv,
- domain,
- ip, ip.clt, ip.dst, ip.src, ip.srv,
- mac, mac.clt, mac.dst, mac.src, mac.srv,
- os, os.clt, os.srv,
- port.srv,
- proto,
- vlan,
- zone, zone.clt, zone.dst, zone.src, zone.srv

Use `.clt` and `.srv` suffixes for *Client* and *Server* in the *Application* Universe which is in client/server mode. Use `.src` and `.dst` suffixes for *Source* and *Destination* in the *network* Universe which is in source/destination mode.

Here are some examples of valid expressions:

- `(ip=10.10.*.* or ip.srv=10.20.30.*) and os.clt='linux'`
- `zone in '/Private/Servers' or port.srv < 1024`
- `(proto=udp and port.srv=53) or zone in '/Private/DNS'`
- `domain='~^www.google.(fr|com)$'`
- `app='http' or app='https'`

Note: `zone = '/Private'` selects only the flows with a client or server zone witch is exactly `/Private` and no other zones.

`zone in '/Private'` select `/Private` and any of its subzones.

CHAPTER

THREE

MAIN TERMS AND CONCEPTS

3.1 General Conventions

3.1.1 Byte metric unit

All byte metric values are given in Byte as `KiB`, `MiB`, `GiB`, etc. As recommended by the INTERNATIONAL ELECTROTECHNICAL COMMISSION (IEC) in 2000 when using power of `2^10` multiple. This means that the values in `MiB` and `KiB` are in binary and equal to 1024 raised to the power of `2` and `1024` raised to the power of `1`, respectively. This notation was designed to distinguish `10^3` bytes (referred as `KB`) and `1024` bytes (referred as `KiB`).

In other words, you would say:

- in decimal notation: `1000` = `1k` (kilo) and `1000^2` = `1M` (mega)
- in binary: `1024` = `1Ki` (kibi) and `1024^2` = `1Mi` (mebi)

For more information about binary prefix, please refer to Wikipedia page (http://en.wikipedia.org/wiki/Binary_prefix).

3.2 Zones

3.2.1 Principles

A *zone* is an **arbitrary container** in which groups of peers can be kept and organized according to their network address.

Each peer being attributed a zone, a conversation between two peers comes with two zones: a client and a server zone.

A zone consists merely of a name, a priority and a set of optional filters. Each conversation is tagged with a client and server zone (using the client and server IP and MAC addresses) according to this process: every rule is tried in order of priority, and the first zone that has filters that comply with this conversation is selected. Thus, it may be important to consider the priority of a zone in the rare occurrence where the default ordering scheme does not yield the expected results.

For instance, here is a simple configuration (in order of priority):

Priority	Name	Subnet	MAC	VLAN	Poller	Device
20	/LAN/Servers/Mail	192.168.1.25		120	localhost	
20	/LAN/Servers/Web	192.168.1.80		120	localhost	
10	/LAN/Servers/Fallback	192.168.1.1-192.168.1.100		120	localhost	
0	/LAN/Fallback	192.168.1.0/24		120	localhost	
0	/Remote				poller2	
-1000	/Internet					

Here, we have two servers (for mail and web) that are tested first by IP (if the VLAN is 120 and the poller if *localhost*), then all other servers (using an IP range), then the LAN, then the remote site (everything from `poller2`), and everything else in `Internet`.

Notice that some fields are unused (MAC, Device), meaning any value will do.

Whatever changes are made in the zone tree, a global fallback (here, it's `/Internet`) will be created by default to store any conversation that is not matched by any rule (this remains true even after filters are added for this zone). Also, this zone is special in that the IP addresses of these conversations will be degraded over time to reduce storage requirements.

Your actual configuration will, of course, be much more complex. Indeed, even the default configuration is larger:

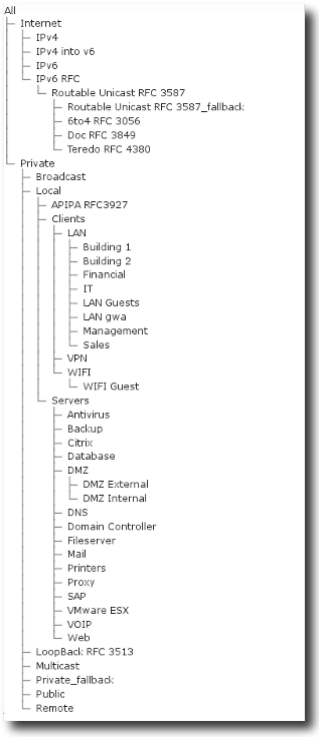


Figure 3.1: Zone tree as displayed in SPV select boxes, showing the default configuration.

3.2.2 Selections

Zone names, although not used in the aforementioned process, play an important role in the GUI. As you can see on the example, zone names are organised in a tree of sub-names delimited with slashes (/), not unlike a standard file system.

For instance, `/LAN/Servers/Web` is made of three components, meant to be read as the host `Web`, amidst the `Servers` in the `LAN`. Here `/LAN` is said to be the parent zone of `/LAN/Servers` and `/LAN/Fallback`, and `/LAN/Servers` is said to be the parent zone of `/LAN/Servers/Mail` and `/LAN/Servers/Web`.

In all select boxes of the GUI, selecting a parent zone will select all conversations that fall in this zone or in any of its child zone.

For instance, in the above example, selecting `/LAN/Servers` will select all conversations in `/LAN/Servers/Mail`, `/LAN/Servers/Web` and `/LAN/Servers/Fallback`.

3.2.3 Fallbacks

By convention, a *fallback* is a zone with a larger filter but lower priority than a set of more specific rules. For instance, in the above example, the `/LAN/Servers/Fallback` zone collects all IP addresses in the 192.168.1.0/24 subnet after some more precise zones tried to match with subsets of this subnet.

Notice that the priority of the fallback must be lower than the priority of these smaller zones; otherwise, they would be shadowed by the fallback.

Notice also that if the example configuration was instead:

Priority	Name	Subnet	MAC	VLan	Poller	Device
20	/LAN/Servers/Mail	192.168.1.25		120	localhost	
20	/LAN/Servers/Web	192.168.1.80		120	localhost	
10	/LAN/Servers	192.168.1.1-192.168.1.100		120	localhost	

i.e., with `/LAN/Servers` instead of `/LAN/Servers/Fallback`, then selecting the `/LAN/Servers` zone in the GUI would actually select `/LAN/Servers/Mail` and `/LAN/Servers/Web` in addition to the fallback. In other words, there would be no way to select in the GUI only the peers that are in the servers IP range but that are neither the mail nor the web server. Using the `Fallback` naming convention allow one to select either a specific server (`/LAN/Servers/Mail`, `/LAN/Servers/Web`), all servers (`/LAN/Servers`) or only the other servers than mail and web (`/LAN/Servers/Fallback`).

3.3 Application

The main objective of *application* is to easily categorize network usage. Through this concept, which is a key notion of Performance Vision, the administrator can group similar network usages into categories that will make sense for his network context. Additionally, by configuring Applications, reports on network traffic are made clearer and are readable by any user regardless of their understanding of the underlying infrastructure (IP addresses and *subnet*, or ports used by each application).

An *application* is a set of network services which together correspond to a business application. For example, an application named *ERP* could be configured to match network traffic on port TCP/80 on a server *Zone* containing the specific server `192.168.20.4/32`.

3.3.1 Application definition

An *application* can be defined using a set of filters a flow must match to enter the application. These filters can use various elements of a flow, from its IP addresses to its ports, poller, protocols, and so on.

Notice that depending on what flow is considered, some of the information may not be available. For instance, the attribution of an application for a NetFlow cannot use anything beside bare IP addresses, protocol and ports. As a consequence, an application defined on a given VLAN, MAC address or protocol stack will never accept a NetFlow.

All rules are checked one after the other and the first matching rule gives the flow its application, in a process similar to the one used for zone attribution.

The priority of these rules can be changed to alter the order in which these checks are performed.

For more information about the configuration of applications, refer to the *Configuration* section.

3.3.2 Examples

An application which is run on a server which has an IP of `192.168.1.4` with *MSSQL* will be defined as follows:

- Port Range: `1433`.
- IP protocol: `TCP`.

- IP Server: `192.168.1.4/32`.

An HTTP application running on a server along with several other applications will be defined as follows:

- Web Application Pattern: `*intranet.securactive.lan*`.

3.4 IP Merging

In order to maximize usage of the available disk space, some information are removed to allow better aggregation. This is the case for IP data of foreign host on aggregation levels 3 and 4.

3.4.1 Principle

Upon data consolidation at the third aggregation level, all IP tagged on the *Internet* zone (or whatever name was given to this default zone) will be removed in favor of a *merged* identifier. Consequently, these IPs will appear as merged in all tables where IP values are displayed if the IP was belonging to Internet Zone and your observation period is such that the third or the fourth aggregation level is used. This will happen with long observation periods (> 8 hours) and also on old data (> 1 week old).

3.4.2 Example

Let's say a user has access to the *Internet* zone using the same application; for example, a web browser using `HTTP` on port `80` to access to different web sites for a period of time. Originally, you will see for that period.

Client Zone	Client IP	Server Zone	Server IP	Application	Traffic
Internet	86.71.197.86	Private (fallback)	192.168.50.34	http	535Bytes

Figure 3.2: TCP conversation before degradation

Once data has been aggregated, if you query the same period of back in time, you will have:

Client Zone	Client IP	Server Zone	Server IP	Application	Traffic
Internet	Merged	Private (fallback)	192.168.50.34	http	5.5KiB

Figure 3.3: TCP conversation after degradation

For the *Client IP*, *merged* means that the two conversations to the different Internet clients have been merged into one single entry. This is only done when the Zone is *Internet* and matches the same server / application couple. So, you still know that this server was accessed from the *Internet* zone with the `http` application on the port `80`.

3.5 Conversation

3.5.1 Objective & Definition

The objective of a *conversation* is to group a set of data exchanges between two hosts for a single *application* into one basic entity to be able to generate a more user-friendly report on network traffic.

A *flow* is a group of data exchanges between two hosts for one *application* over the *aggregation period*. A *conversation* is a group of flows over the observation period. The observation period is defined by a start time and an end time provided by the user. A *conversation* is defined by the following criteria:

- The *device identifier* that received the packets
- The VLAN tag that might be present in the packets

- Source or client IP address (please refer to the chapter *Types of Conversations*).
- Destination or server IP address
- Application (please refer to the chapter *Application*)

3.5.2 Types of Conversations

Performance Vision offers two ways to analyse network *conversation*. From a user's perspective, network conversations can be seen in two different ways, which correspond to two different needs: Client/Server or Source/Destination. This chapter explains how those views differ, which kind of information they provide, and how they can be used.

Source / Destination

In a source/destination *conversation*, all flows between two hosts will be classified following the concepts of source and destination. This means that the flows will group data exchanges from a source IP address to a destination IP address regardless of whether they function as a client or a server.

For instance, a traffic `from A to B` for an application will be broken down in two conversations: a conversation `from A to B` and a conversation `from B to A`.

Src/Dst conversations correspond to a view of network flows for traffic analysis. When reviewing data for traffic analysis purposes, an administrator wants to view flows without considering the role of each host, that is to say, **disregarding if the host is a client or a server**.

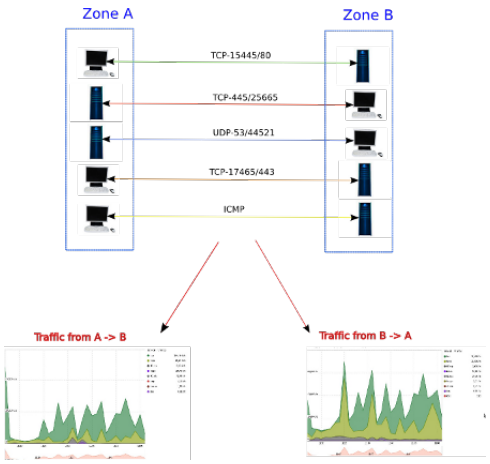


Figure 3.4: Source/Destination treatment

For example, traffic `from A to B` takes into account all traffic coming from a host in A to a host in B, regardless of the role they played (client or server). The above graphs take into account the communications `from A to B`, only in one direction.

Client / Server

In a client/server *conversation*, all flows between two hosts will be classified following the concepts of client and server. This means that the flows will group data exchanges to (and from) a client IP address from (and to) a server IP address.

For instance, a traffic `from A to B` for an application (provided both A and B can be a server for a single application) will be broken down in two conversations: a conversation for `client A & server B` (with traffic `from A to B` and `from B to A`) and a conversation from client B to server A (with traffic `from A to B` and `from B to A`).

Clt/Srv corresponds to a view of network flows for performance analysis. When reviewing data for performance analysis purposes, an administrator wants to view flows **in function of the role of each host, client or server**. Indeed, the role of a host has an impact on the metrics displayed and the clients and servers cannot be mixed.

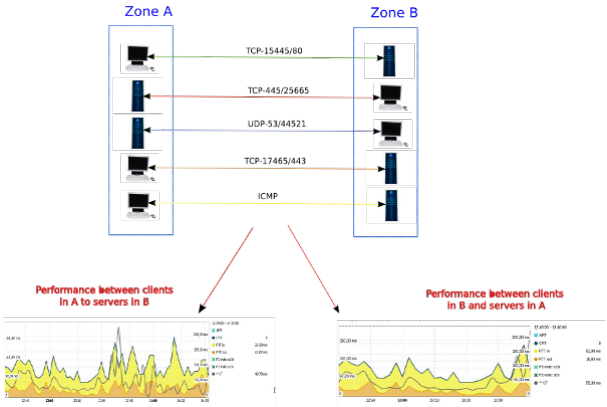


Figure 3.5: Client/Server treatment

For example, the clt/srv graphs shown above will be generated taking into account the communications:

- from clients in A to servers in B
- from servers in B to clients in A

In short, the traffic displayed in client/server conversations will take into consideration the data transfer in both directions.

Note: The appliances can only distinguish reliably clients from servers when the IP protocol in use is TCP, when the connection establishment was successfully received by the probe, and when the connection state is sufficiently active to not be in timeout. In all other cases, the probe assumes that the lower port is used on the server's side.

Where are both being used?

Src/Dst will be used for all views of oriented traffic, i.e., where the reports need to show the amount of data from one zone to another zone. Hereunder (in the first and second lines of the table) you can see that the data exchange between the two hosts has been split into two conversations from A to B and from B to A.

Source IP	Destination Zone	Destination IP	Application	Traffic	Payload	Packets
192.168.80.22	VLAN_Sales (fallback)	192.168.20.208	Smtp-ssl	3.5MiB	3.0MiB	8389
192.168.20.208	VLAN_Labo (fallback)	192.168.80.22	Smtp-ssl	1.2MiB	799.8KiB	7907
192.168.20.237	VLAN_R&D	192.168.10.9	ssh	1.1MiB	771.3KiB	5294
192.168.80.22	VLAN_Sales (fallback)	192.168.20.217	Smtp-ssl	864.5KiB	824.8KiB	752
192.168.80.6	VLAN_R&D	192.168.10.6	Web Intranet Sec...	553.6KiB	521.5KiB	497
192.168.10.9	VLAN_Sales (fallback)	192.168.20.237	ssh	372.1KiB	41.0KiB	5128
204.14.234.36	VLAN_Sales (fallback)	192.168.20.213	Salesforce	352.7KiB	337.0KiB	298
192.168.20.217	VLAN_Labo (fallback)	192.168.80.22	Smtp-ssl	243.0KiB	216.7KiB	498

Figure 3.6: Source/Destination conversations

On the other hand, client/server conversations will be used for all views reporting performance. Hereunder you can see (in the first line of the table) that a client/server conversation takes into account the traffic in both directions.

Client Zone	Client IP	Server Zone	Server IP	Application	Traffic	Packets	Handshake	Transactions
VLAN_R&D	192.168.10.5	Internet	128.237.157.136	Imcu	1.7KiB	18	0	1
VLAN_Sales (fallback)	192.168.20.217	Internet	174.36.30.4	http	3.6KiB	22	4	7
VLAN_R&D	192.168.10.10	Internet	209.85.137.125	NC tcp	2.6KiB	34	4	1
VLAN_R&D	192.168.10.8	Internet	208.71.169.36	Imcu	1.6KiB	17	0	2
VLAN_Sales (fallback)	192.168.20.202	Internet	91.121.2.221	vpn	16.9KiB	184	4	42
VLAN_R&D	192.168.10.4	Mother2	88.191.105.6	Srv_Mother2	16.8KiB	180	4	33
VLAN_R&D	192.168.10.6	Internet	140.211.15.34	http	12.5KiB	31	3	1
VLAN_R&D	192.168.10.4	Internet	193.40.186.4	ssh	606.6KiB	919	4	11

Figure 3.7: Client/Server conversations

In general, you will find that:

- Client/Server is relevant when we are speaking about Performance;

- Source/Destination is relevant when we are speaking about Usage.

3.5.3 Top-Down Analysis

The Src/Dst matrix can be the starting point for a fine-tuning analysis of traffic: bandwidth and conversation. In each cell, there are two buttons:

- one to display the bandwidth graph `from zone A to zone B`
- one to display the conversations `from zone A to zone B`.

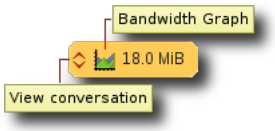


Figure 3.8: Cell detailed view

The first link will open the conversation table and will display all the traffic between the two zones, whereas the second one will display a bandwidth chart from the source zone on the left and the destination zone on the top.

3.6 Data Aggregation

3.6.1 Rationale

By nature, the operations of statistical analysis performed require the storage of large amounts of data. Furthermore, that data must be stored over extended lengths of time so as to expose overall trends. In order to minimize storage space while still making it possible to reveal trends over weeks or months, Performance Vision automatically summarizes the collected data. The process of creating these summaries is called aggregation.

3.6.2 Process

Aggregation occurs automatically. Whenever your probe displays a chart or a table, this is based on already aggregated data. In order to display this, Performance Vision first decides on an aggregation granularity depending on the length of the time period you requested and how far back into the past it goes.

Aggregation granularity	Storage duration	Request length for tables	Request length for graphs
2 minutes	48 hours	60 minutes	120 minutes
15 minutes	7 days	8 hours	16 hours
2 hours	2 months	2 days	5.25 days
1 day	1 year	359 days	359 days

For example, with graphs, if you want a data granularity of two minutes, you can request a period length up to 120 minutes anywhere during the last two days.

With tables, if you want a data granularity of two hours, you can request a period length up to two days anywhere during the last two months.

Note that because the larger aggregate levels summarize more data at once, they take up less disk space, and can be kept in storage much longer without filling out the hard drive. This strikes a good balance between data granularity and duration of retention: performance data for the last two days is available with the best granularity, and long-lasting global trends can be exposed from as far back as one year (albeit with less detail), all from the same interface.

Aggregated data is computed, in a nutshell, by identifying network conversations where the same server and the same client talked using the same application, and grouping them together. The metrics for each such group are summed up in accordance with their mathematical nature (for instance, packet counts are added and response

times are averaged per packet), so only one line of data is retained for each conversation group. This line still contains a relevant summary of your network and application performance, but it's storage takes up a lot less disk space.

Example: A user checks out a Web page once at `16:38`...

Figure 3.9: Flow example at `16:38` to `16:40`

... and again at `16:41`.

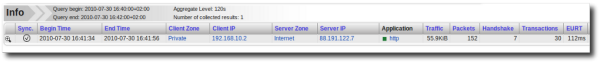


Figure 3.10: Flow example at `16:40` to `16:42`

Here is the aggregated line for both events if you query between `16:38` and `16:42`:

Figure 3.11: Flow aggregation from `16:38` to `16:42`

Observe that the traffic, and the packet, handshake and transaction counts have been added, and the EURT averaged. For example, the *handshake* is now 19 `(12 + 7)`.

Note: Performance Vision requires a complete set of data for an aggregate level to compute its summary. This is the reason why captured network events don't appear right away on your probe. The probe first waits until the end of the minimal aggregate time of 2 minutes, computes its summary, and only then is the aggregated data for these last 2 minutes made available in the interface.

CHAPTER

FOUR

METRICS COMPUTATION

Here, you can find details on how some of the less obvious metrics are computed, and how they are affected by the sniffer configuration. You may safely skip this section unless you need a deeper understanding of how the sniffer works.

4.1 Conversations

Many generic metrics are computed on TCP streams. To be able to interpret these correctly, it may be useful to be aware of a few things.

4.1.1 Client or Server?

To find out which peer is the client, the sniffer tries several options:

- If it understands the protocol at hand (and has successfully identified it), then client/server identification is usually trivial. Unfortunately, most traffic does not fall into this category.
- In TCP, the client is the peer that actively opens the connection (i.e., sends the initial SYN). But we may miss the SYN or we may have forgotten it if we have not received traffic for that socket for more than 2 minutes (especially problematic for lengthy connections such as remote control protocols).
- In either TCP or UDP, we may have indicative port numbers. A port number below 1024 on one side and greater than 1024 on the other is a strong indication of the server location.
- In TCP, we may have seen past SYNs directed at one of the ports, which again gives an indication of that port being the server.
- When all else fails, the server is chosen according to a complex heuristic that's mostly equivalent to choosing at random.

4.1.2 Keep-Alive

Applicative keep-alives are small messages that are sent from either peer to the other when no traffic have used this socket for some time. They must not be taken into consideration when computing *SRT*, *DTT*, and so on. The `ica_keepalive_max_size` parameter is dedicated to the detection of ICA (citrix) keep alive messages.

The standard TCP keep-alive packet is normally detected using its size and sequence number, according to the RFC. In case the previous sequence number is unknown, though, the `tcp_keepalive_timer` may be used as an alternative: after this inactivity period, any TCP packet that looks like a Keep-Alive will be ignored.

4.1.3 DTT timeouts

The objective of the TCP DTT metric is to measure the duration of a single write (or of a sequence of closely related writes). For protocols that do not follow the pattern request/response, it is very important to detect when two data transfers are separate in time (suggesting they are unrelated). The `tcp_dtt_timeout` parameter helps

with that. If two packets are separated by more than this duration, then they do not belong to the same DTT. By default, it is set to 1s so that lost packets nor a full reception buffer would not interrupt the DTT, but an actual pause from the sending application will be detected as such.

4.1.4 What is a retransmission?

According to the sniffer, any TCP packet with a payload (or a SYN, a FIN or RST flag) which a sequence number that was already covered is a retransmission (here, *covered* means that this sequence number was in a packet that has already been analyzed).

Fast retransmission is thus counted as retransmission.

4.2 HTTP

The HTTP metric offers a very synthetic notion of a *page*, which is a set of HTTP documents fetched by the same user and combined by his browser into a single object, a "page". Reconstructing pages from the actual packets involves an unusually high number of operations and thus, deserves quite a detailed description.

4.2.1 HTTP specific glossary

Although not required to use Performance Vision, the following definitions are required to understand the following description.

- HTTP message: as defined by RFC, it is an HTTP header optionally followed by a body. Sniffing gives us some of the headers, the relevant timestamps, sizes, and so on. We may not see everything, but the beginning of the header is mandatory in order to recognize an HTTP message.
- HTTP query: HTTP message with a command (GET, POST, HEAD, etc) and the URL.
- HTTP response: HTTP message with a response code (sometimes called *status code* or *status*)
- HTTP hit or transaction: HTTP query with optionally its associated HTTP response (note: a response with no associated query is ignored for this metric).
- user: the HTTP client software (browser or whatever) that has sent the query under consideration. It's identified by his IP address and user-agent field.
- page: set of transactions that are supposed to be perceived as a single query implying a single delay for the user. Notice how subjective this definition is. The intent is to include in a single page all the hits required for a typical browser to display enough content for the typical user to think his query is fulfilled. For websites or browsers that delay download of content until it becomes visible, or for websites that display intermediary content, the only objective is to behave in a way that's understandable.
- root (of a page): the transaction that triggers other transactions for the same page, either directly or indirectly. We'd like it to be the first chronologically but that's not necessarily the case due to mirroring.

4.2.2 From packets to HTTP messages

The sniffer receives fragments of HTTP messages. It starts to reconstruct a new HTTP message as soon as it receives the start of a header. Some fragments of the message may be missing, though, in which case it may be incapable of:

- associating a body fragment to the proper HTTP message, thus leading to erroneous payloads and dubious chronology,
- saving part of content in HTTP save files (without notice),
- reporting the timestamp of message end.

4.2.3 From individual messages to transactions

HTTP offers no better way to associate response with corresponding query than to rely on ordering: first response of the socket with first query, and so on.

So, for every socket, the sniffer stores all queries not already paired with a response. Notice that on a socket, a proxy may mix queries of different users, and that two interconnected proxies may even mix queries to distinct servers.

Notice also how damaging a single dropped packet may be if it hides a query or a full response to the sniffer, since all pairing following this gap will be questionable.

Also, servers may not respond, leading to a timeout of the pending queries (which will be inserted in database without any response).

4.2.4 From transactions to pages

Since all transactions of a page are necessarily emitted by the same user, then all transactions are associated to this user, in chronological order (time and the "Referrer" field are our two best tools from now on). Notice that since a page routinely involves transactions of several sockets, and since that different sockets are reassembled by different TCP parsers which thus delivers segments at different pace, then it's possible for the HTTP metric to reconstruct a transaction A before a transaction B even if B happened and was received by the probe before A (for instance, if A's socket reassembly was delayed by a missing frame). In such occurrence the referrer relation between A and B may not be honored.

We do not wait for the pairing with a response to attach a query to the page it belongs to. When we attach a new query to a client we look for the referrer of this transaction within the ones that are already attached to this client (in case the referrer field is absent we use the same kind of referrer cache as found in KSniffer (https://www.usenix.org/legacy/event/osdi04/tech/full_papers/olshefski/olshefski.pdf)). If the referred page is itself attached to another page two behaviors are available:

- we detach it, thus turning the referrer into the root of a new page
- or we follow the chain of attachment and attach the new transaction to the parent page

Note that the first behavior is possible only when the content-type of the referred page does no prevent it (ie. is not typically reserved to non-root transactions, such as image, css, and other typically embedded content).

You can choose between these two behavior with the `http-detach-referred` parameter.

Second behavior (keep referred transactions attached) is better when iframes are involved but it is believed that the first (and default) one should generally leads to better results (other than iframes, the only observed case where a referenced transaction was obviously not a page root was an ajax POSTing to the same URL as referrer continuously, thus detaching it's predecessor).

If/When we eventually receive the response of a transaction (and thus, hopefully, its content-type) we revise our judgment on the attachment: if the transaction seems to have not been triggered by AJAX, and its content-type is indicative of a standalone document (pdf, ps or html with status 200), then we detach it (turning it into a root). Otherwise, if the content-type is not indicative of a typically embedded content (image, css, etc) then we check the delay between the page root and this transaction and if found greater than a parameter (`http-page-construction-max-delay`) then its detached as well.

To speed up information retrieval some global per page values are precomputed in the sniffer: every transactions attached to a page contribute its counters into the page, as soon as it was received less than `http-page-contribution-max-delay` seconds after the root. All these transactions will contribute to the page load time.

To be able to dump a root transaction with all these counters we must of course delay the dump of roots as much as possible, thus raising memory requirements.

4.2.5 Protections

To limit memory and CPU usage the sniffer implements these protections:

- page reconstruction is only active for some IP addresses and TCP ports (client or server). See the HTTP flag in zone and applications definition. All transactions that do not comes from / goes to one of these IP addresses will not be attached to a root transaction. It will be inserted in database but will be excluded from page list.
- the total number of simultaneously tracked and remembered HTTP transactions is limited by `http-max-tracked` (unlimited by default). New transactions above this will be ignored (with catastrophic consequences on transaction pairing).
- the total number of simultaneously tracked and remembered HTTP transactions for which we want page reconstruction is limited by `http-max-tracked-for-reconstruction` (unlimited by default).
- max size of http save file is limited by `http-max-content-size` (50k by default).
- the memory dedicated to the referrer cache is limited by `http-referrer-mem`.

4.2.6 Limitations

Page load time is the most interesting metric, yet we have seen that many conditions must be met to accurately reconstruct pages.

- the process is very sensible to missing TCP fragments (retransmitted fragments cause no problem but fragments that are not mirrored to the probe do)
- the bigger the proxies, the less reliable client isolation will be
- some heuristics regarding AJAX, content types and timing does not necessarily match your sites
- some client may successfully hide the referrer (or worse: we may guess a wrong referrer)
- HTTP analysis may consumes more resources than what's available (or configured)
- any small inaccuracy in HTTP message reassembly or in transaction pairing will lead to much bigger inaccuracy of page load time.

CHAPTER

FIVE

DEPLOYMENT

5.1 How to integrate Performance Vision in your network?

5.1.1 Preliminary steps

Performance Vision is dedicated to analyzing the performance of business critical applications in a corporate network.Hence the very first step before considering integrating Performance Vision in your network, is:

- identifying an up-to-date list of business critical applications (including applications directly supporting business processes, but also applications on which these may rely – e.g. DNS, Microsoft-DS etc...).
- locating the servers hosting these applications.
- defining which network devices clients are using to access these applications.

5.1.2 Positioning the probe

Performance Vision appliance will be installed as close as possible to to the servers to provide the best analysis. Measurements are more accurate if the probe is located in a central location next to the server and you will get a wider view on the performance experienced by all the users connecting to this server.

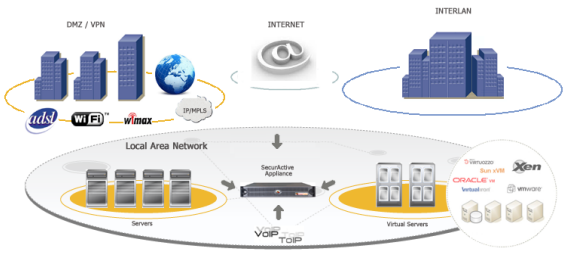


Figure 5.1: SPV network positioning synoptic

5.1.3 Choosing a traffic capture method

Two main methods may be used to establish a permanent point of traffic capture: `TAP` or `SPAN`. A `TAP` is a network device which will installed in-line on the network and will send a copy of the traffic on one or two listening ports of the probe. A `SPAN` (also commonly called port mirroring) is a feature of network switches that enables a network administrator to send a copy of a given traffic (on one or several interfaces / VLANs to a mirroring port).

The most commonly used method is the `SPAN` port (port mirroring) mainly because it enables administrators to monitor potentially any traffic going through the switch, with an existing network device. Collecting traffic through a `SPAN` port will likely not generate any additional point of failure on the network and will be regarded as a minor modification of its existing configuration. Network `TAPs` are also an option (if no `SPAN` is doable for

example) but the traffic captured will be limited to the network link(s) going through the TAP. A connection via TAP induces additional costs.

If you choose to capture network traffic through a SPAN, you should pay a specific attention not to copy twice the same traffic to the listening interface of the probe (which would degrade the statistics provided by the probe).

5.2 How to capture traffic?

Performance Vision can rely on two mechanisms to capture network traffic: Port Mirroring (commonly called SPAN) & TAP (Terminal Access Point).

5.2.1 Port mirroring

Port mirroring, also known as SPAN or *roving analysis*, is a method of monitoring network traffic which forwards a copy of each incoming and/or outgoing packet from one (or several) port(s) (or VLAN) of a switch to another port where the analysis device is connected. Port mirroring can be managed locally or remotely. To configure the port mirroring, an administrator selects one or several ports from which all packets will be copied (source ports) and another port or ports where the copy of the packets will be sent (destination port). The administrator can include either all packets in the port mirroring or only the transmitted/received packets. In case both transmitted and received packets are included, a packet going from a 1st monitored port to another monitored port will be copied twice to the destination port. This will have an impact on the measures and performance provided by the analysis device (e.g. retransmission rates, response times, ...). Performance Vision captures and evaluates the data without any impact on the original traffic.

The port mirroring is the most commonly used solution to capture traffic, because it is inexpensive, flexible in terms of how much traffic can be captured at once and remotely configurable.

Please note that a port mirroring may have some drawbacks, such as:

- It can consume significant CPU resources while active
- There is a risk of not receiving some packets (like media errors)
- In the case of traffic congestion at the switch level, the port mirroring is likely to drop some traffic (because the SPAN process does not have priority).

In some cases, a better solution for long-term monitoring may be a passive TAP or an Ethernet repeater ("hub").

Advantages

- Low cost (this feature is embedded in most switches)
- Can be configured remotely through IP or Console port
- The only way to capture intra-switch traffic
- A good way to capture traffic on several ports at once

Drawbacks

- Not adequate for fully utilized full-duplex links (packets may be dropped)
- Filters out physical errors
- Impact on the switch's CPU
- Can alter the timing of the frame (with an impact on response time analysis)
- SPAN has a lesser priority than port to port data transfer

5.2.2 Network TAP

A network TAP (Terminal Access Point) is a hardware device which can passively capture traffic on a network. It is commonly used to monitor the network traffic between two points in the network. If the network between these two points consists of a physical cable, a network TAP may be the best way to capture traffic. The network TAP has at least three ports: a port A, a port B, and a monitor port. To place a tap between points A and B, the network cable between point A and point B is replaced with a pair of cables, one going to the TAP's A port, one going to the TAP's B port. The TAP passes all traffic between the two network points, so they are still connected to each other. The TAP also copies the traffic to its monitor port, thus enabling an analysis device to listen. Network TAPs are commonly used by monitoring and collection devices. TAPs can also be used in security applications because they are non-obtrusive, are not detectable on the network, can deal with full-duplex and non-shared networks, and will usually pass-through traffic even if the tap stops working or loses power.

Advantages

- No risk of dropped packets
- Monitoring of all packets (including hardware errors -MAC & media)
- Provides full visibility including congestion situations

Drawbacks

- The device may require two listening interfaces on the analysis device
- Costly
- No visibility on intra-switch traffic
- Not appropriate for the observation of a narrow traffic range.

5.3 Supported Protocols

The SPV sniffer can detect all Ethernet packets even if those packets have a VLAN tag in their Ethernet header. SPV also accepts both IPv4 and IPv6 protocols.

Note: Non Ethernet flows are invisible for the SPV solution.

5.3.1 Non IP Protocols

If the Ethernet protocol is not an IP protocol, it will appear in *Non IP* submenu. All those data will not appear elsewhere.

Figure 5.2: Non IP protocols menu

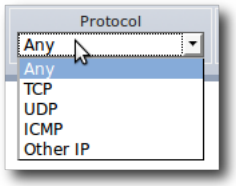


Figure 5.3: Level 3/4 protocol filter

5.3.2 IP Protocols

Ipv4 and IPv6 are both captured and splitted in four *Level 3/4* protocols: TCP, UDP, ICMP and OtherIP.

Some of those data are duplicated in other specialised categories: *Web*, *VoIP*, *DNS* to display more specific metrics.

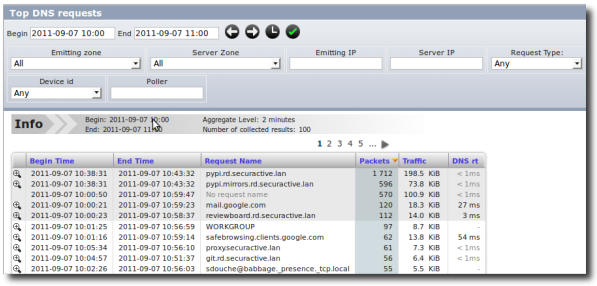


Figure 5.4: DNS specialied view

5.3.3 Limitations

If the rate of incoming packets exceeds the rate at which the sniffer can parse the traffic for too long then some packets may be dropped by the Linux kernel. These packets won't get accounted for in the GUI.

As a realtime protocol analyzer, the sniffer is also limited in what protocols it supports and how deep it inspects packets. Here is a quick overview of the most blatant limitations:

- Ethernet parser supports Linux cooked capture extension (used when capturing on "any" interfaces) and 802.1q vlan tags. All other Ethernet extensions are ignored.
- ARP parser knows only Ethernet and IP addresses.
- DNS parser support MDNS, NBNS and LLMNR in the extend where these protocols mimic legacy DNS (with the exception that it can unscramble NetBios encoded names).
- FTP connection tracking merely look for PASSV or PORT commands in the TCP stream without much care for the actual protocol.
- TCP options are ignored.
- Postgresql parser supports only protocol version 3.0 and Mysql parser supports only protocol version 10. This should cover most of the installed base, though.
- TNS parser (for Oracle databases) was roughly reverse engineered from various sources, especially the wireshark source code. It should thus not be expected to understand all messages in all situations.
- SIP parser implements no proprietary extensions, however prevalent.
- As there are no concept of connections for UDP, UDP conversations are ended after a timeout period of 2 minutes without any packet in any direction. This might not match the underlying protocol.

- VoIP dialogs are identified by their call-id only, which imply that if the sniffer listens to various independent SIP proxys or servers then call-id collisions can not be ruled out (this choice was made because it proven useful in practice).

5.4 Port-mirroring and duplicated packets

5.4.1 Introduction

The configuration of a port-mirroring session has to respect some specific rules and standards. The main goals of a port-mirroring session are to:

- Gain insight into the highest number of flows, which are seen as strategic by the IT manager
- And ensure that all collected flows are appropriately analysed.

It is crucial to ensure that a minimum number of flows are duplicated to the interfaces.

5.4.2 Detail

SPV solution can manage any level of traffic duplication (dropping packets received in excess) ; this, however, involves a significant loss of performance. There are two main rules:

- Basic port-mirroring sessions, also called **1 to 1** port-mirroring session. This configuration does not generate duplicated packets. However, increasing the number of **1 to 1** port-mirroring sessions could produce this phenomenon.

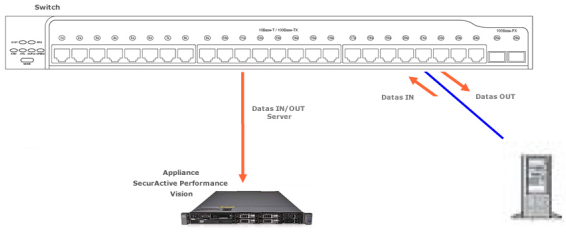


Figure 5.5: “1-to-1” port mirroring session

- Multiple port-mirroring sessions, also called **N to 1** port-mirroring session. In this specific event, the duplicated packets phenomenon can occur.

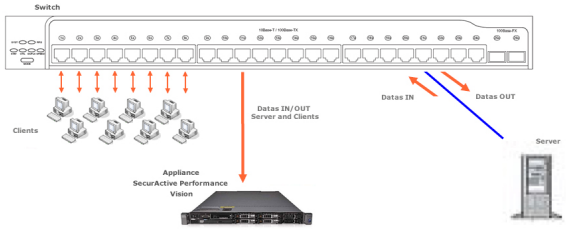


Figure 5.6: “N-to-1” port mirroring

Warning:
- According to the number of listening points, in a multi-switch mode this phenomenon can occur despite the use of a **1 to 1** port-mirroring session.
- A VLAN is a definition of a set of ports; this means that the port-mirroring session is a **N to 1** port-mirroring session.

5.4.3 Some examples of duplicated packets / non-duplicated packets

In a standard port-mirroring configuration **(N to 1)**, it is highly likely that some transmitted packets to the appliance are duplicated. In the following example, configuring a port-mirroring session on both the IN traffic and the OUT traffic of the switch means that the appliance will receive twice the same traffic:

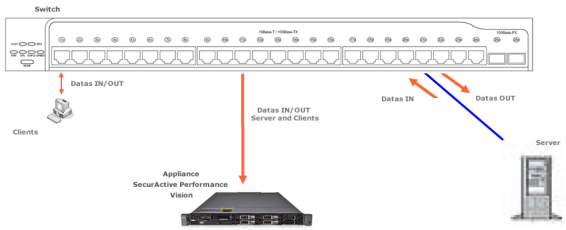


Figure 5.7: Example with duplicated packets

By only listening to the IN traffic (or only the OUT traffic) on the Ethernet ports concerned, we will ensure the flow transmission to be in a unique way for the sessions between the client and server, thus avoiding the duplication of packets:

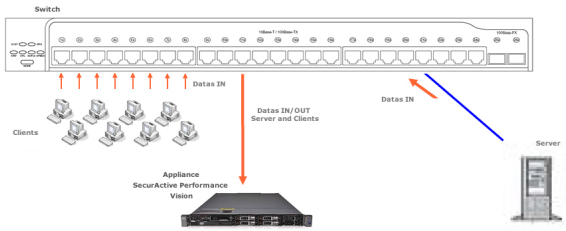


Figure 5.8: Example without duplicated packets

Note: In the event of a **N to 1** port-mirroring session, the total bandwidth of the "source" Ethernet ports of the mirror should not exceed the maximum bandwidth of the "destination" Ethernet ports of the mirror.

5.4.4 Removal of duplicated packets

The SecurActive system checks and controls the duplicated packets phenomenon on all listening ports. It also ensures all duplicated packets are removed. However, in some cases, some duplicated packets could be mixed up with retransmitted packets.

It is therefore crucial to minimize the duplicated packet rate (or at least to arrange the mirroring such that duplicates follow the original as closely as possible). In order to reach a low rate of duplicated packets, the appliance provides information on the duplicated packet rate though the Pulsar command:

```
Welcome to Pulsar - the SPV shell - v1.13.0

Type `help` to display Pulsar commands.
Type `help COMMAND` to display the command help details.

vsonde73# analyzer mirror
5.12%
vsonde73#
```

Figure 5.9: Information on the duplicated packets rate in Pulsar

This means that `5.12%` of the listening traffic is duplicated.

5.4.5 Deduplication algorithm

The sniffer usually receive frames from multiple locations on a network, and so it can be cumbersome (if not impossible) to avoid the situation where the same frames are mirrored several times toward the probe. Deduplication is the process of ignoring selectively packets that are artificial duplicates due to the network infrastructure. On the other hand, automatic deduplication makes it harder to find out if duplicates were present in the network in the first place.

The following chapters covers the deduplication system in order to help minimizing duplication issues.

The packet sniffer detects and drops duplicate frames based on a digest of their *content*, which is compared to the digest of *the packets* received *shortly* before. After this rough description we are going to see in more depth over what content is computed the aforementioned signature, which previous packets are considered and how short the sniffer looks for duplicates in the past.

When computing the digest, only a selected set of bytes are compared:

- For small frames (which size is below the size of an IP header) all bytes are taken into account,
- For bigger frames, bytes after the Ethernet header (including the VLan tag **if collapsing VLans**) and up to the 64th byte of the frame (or less if the frame is smaller), excepting the TOS, TTL and IP checksum fields are taken into account.

The rational behind skipping *Ethernet* header is that we want to pair two packets if only their Ethernet addresses (or VLan tag) differ (one is a copy of the other, merely one switch away from it). The rational behind excluding TOS, TTL and checksum fields of the IP header is to be able to pair two packets when one is a copy of the other, only one hop away from the first one (after traversing one or several routers).

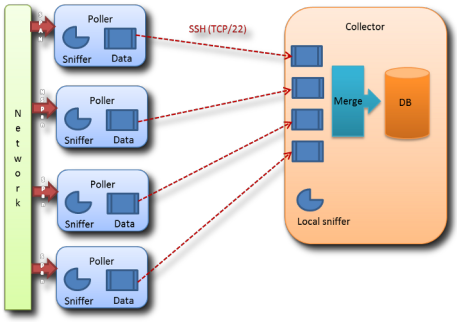
Then a packet digest is build from the remaining bytes and compared to those of previously received packets. If one is found with same signature, the new packet is dropped. If one is found that is older than the older expected duplicate then the packet is allowed to proceed.

The age of the oldest expected duplicate is set by a runtime parameter which default value is 100ms. This default value should fit most settings.

5.5 Distributed Architecture

5.5.1 How does the distributed infrastructure work?

Appliances hosting only the sniffer component of SPV are called “pollers”. The appliance hosting the components in charge of collecting, merging and integrating the data from the pollers into a single database is called “collector”. The collector appliance may also host one sniffer component.



The pollers listen and analyze the network traffic. The collector receives data from the pollers, integrate them in the database, and then provides an access to the data through the Web UI.

You can add a new *poller* via *Pulsar* by using the command `poller add <IP>`. The specified IP of the *poller* must be reachable with `SSH port 22`.

The *Pollers Status* page in the *Configuration* menu display some status information about pollers.

5.5.2 Where is data being merged / segregated?

The data is merged (i.e. the data is integrated in the reports with no consideration for the poller, which has captured it) in:

- Business Critical Application Dashboard.
- Business Critical Network Dashboard.
- Application dashboards.
- Graphs (performance, bandwidth, matrix).
- Comparison tables (Client / Server, Network performance, Application performance).

Please note that in these reports, you can enter a filter to view the data captured by one poller only. The data is segregated (i.e. the data is kept separated depending on the poller which captured the data) in all other tables. [1] Please note that in these reports for a single conversation viewed by two pollers, you will get two lines.

5.5.3 What happens if a poller does not answer?

If a connection to a poller is broken, the collector wait for it during 10 minutes. After this time interval, the collector will flag the poller as 'missing'. After these 10 minutes, the collector stops waiting for the missing poller and restarts its activity. Data integration will be 10 minutes shifted upon missing poller response again. See example bellow:

```
min00
^^^^^
poller1 ok
poller2 ok
=> data integration

min02
^^^^^
poller1 ok
poller2 fail
=> wait for poller2 [min02]

min04
^^^^^
poller1 ok
poller2 fail
=> wait for poller2 [min02, min04]

... same, wait more and more poller2 data ...

min12
^^^^^
poller1 ok
poller2 fail
=> integrate data of poller1 for "min02"
=> wait for poller2 [min04, min06, min 08, min10, min12]

min14
^^^^^
poller1 ok
poller2 ok
=> integrate all data poller1 and poller2

Conclusion
^^^^^^^^^^
Data lost: poller2 [min02]
```

[1] This may never be developed.

5.5.4 How configure a poller?

All pollers are available via SSH using the Pulsar shell, just like you access to the collector (please refer to *Pulsar*). A poller shell allows you to configure the poller IP, hostname, etc. But some commands like `reset` or `poller` are not available.

The collector's shell allows you to show and to create or delete pollers. To do this, please use the `poller` command (`help poller` for details).

5.5.5 Limits

The distributed architecture provided by version 2.5 has some intrinsic limits:

- There is no feature for deduplication between pollers (i.e. a network flow captured by two pollers will counted twice in reports that merge data from several pollers). [2] (But you can filter the data for each poller.)
- If there is some load balancing at the packet level (and not at the session level) and two pollers view two different parts of the traffic, the collector will not be able to rebuild this flow and no performance metric will be available in this case. [3]
- The positioning of each poller with regards to client and server will have some impact on some metrics (SRT, RTT Server, RTT Client, RR Server, RR Client, . . .)
- The maximum number of sessions handled by the collector remains unchanged (approx.. 100k concurring sessions).

5.5.6 Prerequisites

- All pollers have to be synchronized to a single *NTP*.
- All pollers and collector require an administration port connected to the network and a fixed IP address.
- Connectivity between pollers and collector on port `TCP/22` is required.
- Some network capacity is required to transfer teh data from the pollers to the collector (current evaluation is 0.2% of the bandwidth analyzed).

5.5.7 Adequate / non-adequate implementations

Situation	Fit for version 2.5	Comments
Two data-centers (Active / passive)	Distributed may or may not be required.	Most applications will be deployed in normal conditions on DCa; if in normal conditions DCb, receives no production traffic, hence a second probe may not required; if applications are, in normal conditions, distributed between DCa and DCb, then a distributed implementation is required.
Two data-centers (Active / Active)	Distributed is adequate.	If the traffic between servers is captured, it may double counted ; traffic from clients to servers should not be double counted.
N data-centers through WAN.	Distributed is adequate.	Traffic between servers will be captured twice and double counted.
N Datacenters and M remote sites	Distributed may not be adequate.	The traffic going from the remote sites to the datacenters will be double counted. The cost of deploying physical units may be superior to the benefit.

[2] This corresponds to a rare case ; this case is not handled by the non distributed implementation of Performacne Vision, nor by most competitors. The bypass option would be to use TAPs to re-aggregate both flows before it reaches the interface of the poller.

[3] This is already the case in a non distributed implementation. The only new element is the fact that data will be more readable if all pollers have the same capture points.

5.6 Virtual Performance Vision

Note: For more details about step-by-step virtual appliance installation cf *Virtual Appliance Step-by-Step*.

If you are installing the virtual image of Performance Vision then you have a to take into account a few additional facts.

5.6.1 How to get the image

This section is based on version 2.5.13, the filename will evolve depending on the version number.

The ZIP archive will contain the following files:

```
- SPV-2.5.13-r2.mf
- SPV-2.5.13-r2.ovf
- SPV-2.5.13-r2.disk1.vmdk
```

5.6.2 Virtual hosts settings

Performance Vision virtual appliance is designed to run in a VMWare ESX v4 or v5 environment. It can be lounched with a minimum of 512MB of RAM although a larger quantity is recommended to ensure satisfactory performance rates.

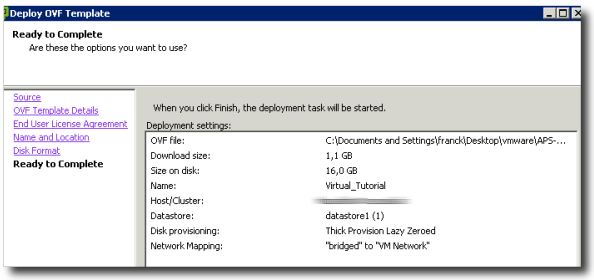
However all settings cannot be tested; in case of doubt it is recommended to fall back on these tested settings:

- RAM: 512MB, 4GB, 6GB, 8GB, 12GB or 16GB;
- CPU: 1, 4 or 8;

5.6.3 Installation

1. Connect to your Vsphere Client and then in the Virtual Machines tab, in the “File” menu, select “Deploy a new OVF template”.
2. Find and open the Performance Vision OVF file.
3. Click on “Next” twice and then accpt the license agreement
4. Name the Virtual Machine appropriately (SPV applicance for example).
5. The system detects the space available on the disk for the new Virtual Machine, we recommend to allocate the following spaces:
   - Trial Virtual Appliance: 4GB RAM, 2 vCPU > 2,0 GHz
   - Virtual Poller: 8 GB, 2 vCPU > 2,0 GHz,
   - Virtual Appliance: > 16 GB, 4 vCPU > 2,4 GHz

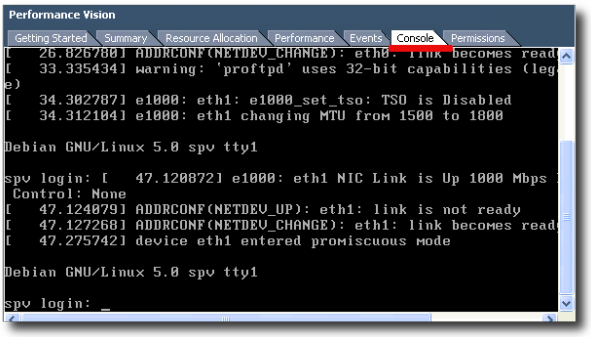
   You get:



6. Click on “Finish”, the Virtual Appliance gets installed. You will get notified when the installation is complete.
7. Once the Virtual Appliance is installed, you have to start it by clicking on “Power on the Virtual Machine” or on the green triangle.

5.6.4 Access the virtual console

Display the Console tab and access the CLI interface named “Pulsar”.


Performance Vision
Getting Started
Summary
Resource Allocation
Performance
Events
Console
Permissions
[26.826780] ADDRCONF(NETDEV_CHANGE): eth0: link becomes read
[33.335434] warning: 'proftpd' uses 32-bit capabilities (leg
e)
[34.302787] e1000: eth1: e1000_set_tso: TSO is Disabled
[34.312104] e1000: eth1 changing MTU from 1500 to 1800
Debian GNU/Linux 5.0 spv tty1
spv login: [47.120872] e1000: eth1 NIC Link is Up 1000 Mbps
Control: None
[47.124079] ADDRCONF(NETDEV_UP): eth1: link is not ready
[47.127268] ADDRCONF(NETDEV_CHANGE): eth1: link becomes read
[47.275742] device eth1 entered promiscuous mode
Debian GNU/Linux 5.0 spv tty1
spv login: _

The probe is launched. When the network interfaces turn into promiscuous mode, click on the Console view and then “Enter” to display the login prompt. Please note: clicking on the black screen deactivates your mouse. To reactivate it, you can use the key combination “Ctrl + Alt”. To configure the probe, please refer to the *Pulsar* chapter. After configuration you have to reboot the virtual applicance.

5.6.5 License

Except the experimental virtual appliances for testing provided from our Web site, the virtual appliances are delivered without license key. You normally receive this key by e-mail at the product’s delivery. If it is not the case, please contact our sales department: sales@securactive.net (sales@securactive.net).

To install a license package (as well as an upgrade package), proceed as usual (see *Licensing and Upgrades*).

5.6.6 Capturing traffic

Virtual appliances are configured with only two network interfaces:

- eth0 for administration
- eth1 for sniffing traffic

Any additional virtual adapters you may add will be listened for traffic by the packet sniffer.

Actual packet capture depends on the virtual switch you are using.

In the realm of VMWare’s bundled Virtual Switch the `promiscuous mode` (beware that name is misleading) is actualy a port mirroring. Also, depending on the virtual switch configuration, if the packet sniffer sets the promiscuous bit of the eth1 virtual adapter, the mirroring mode will be activated automatically. Refer to the Virtual Infrastructure Client manual (http://www.vmware.com) for further details.

Under VMware Player you need to configure eth1 as a bridged device, and give permission to the virtual appliance to turn it into promiscuous mode.

Other virtual switches may have different/more features.

5.6.7 Data storage

Virtual appliances come with no data disk, thus everything (traffic data as well as pcaps and reports) will be written to the system disk only.

If you plan to keep a long history of data then a dedicated data disk is mandatory. To create one, attach a new drive to your VM and then run the `format_data_disk` command from `pulsar`.

Notice that:

- you will not be able to resize this data disk hereafter (the required size depends on the traffic you plan to monitor but anything below 500GB seams dubious);
- the data previously acquired will be lost;
- you are required to reboot the appliance once done.

5.7 Netflow

5.7.1 Overview

Any SPV poller can be sent netflow v5. The pollers will add volumetry informations of every netflow in the traffic statistics so that these flows will be visible from the GUI. In this case, IP address of the sending equipment will be displayed next to the receiving poller name.

5.7.2 Configuration

By default, pollers listen to UDP ports 2055, 9555 and 9995. These ports can be changed in sniffer configuration from the GUI. Clear this list of ports to disable the feature.

Note that the sniffer must be restarted after this change.

5.7.3 Limitation regarding reception

Netflow export, transported by UDP datagram, is a best effort service. A switch may skip sending a flow if it's overloaded. The network may skip forwarding the datagram if it's congested.

To make things worse, SPV does not currently report missing netflow frames.

5.7.4 Limitation regarding content

Instead of the many measurements undertaken from the mirrored packets, netflow provides only mere volumetry; such as, for each IP address, protocol and ports:

- start/stop timestamps
- packets and bytes count
- number of packets
- number of TCP SYNs, FINs, RSTs
- ToS
- switch input/output port numbers

5.7.5 Limitation regarding collection

Netflows are typically exported only after each individual flow is idle for more than a given timeout, grows larger than a configured threshold or is active for more than a given duration. This later parameter is an important concern. If the max age of a netflow is allowed to exceed SPV data integration period (2 mins) then received netflows risk being late for database insertion and ignored.

This is much shorter than most installations. For instance, default activity duration for CISCO equipment is typically 30 minutes.

Attention: Configure all your netflow emitters to expire flows after not (much) more than 2 minutes.

CHAPTER

SIX

VIRTUAL APPLIANCE STEP-BY-STEP

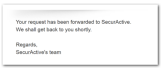
6.1 How to get the image of the Virtual Appliance

You can get a image from the Internet from .

You should reach a subscription page with a form that needs to be filled such as below.

Then complete the Captcha and submit such as below.

You will be notified that your request will be forwarded to Performance Vision



If your request is granted, you'll receive 2 emails from Performance Vision.

First email subject: **Performance Vision Evaluation - Documentation**

Second email subject: **Performance Vision - Download of your evaluation Virtual Appliance**

From that second, click on the **Download Link** which will lead you to a page such as below

You will need to Download this file.

6.2 Virtual Appliance Specifications

The Performance Vision Virtual Appliance is designed to run in a `VMWare ESX` v4 or v5 environment.

It is designed to run with a minimum RAM of `4096MB`, although a larger quantity is recommended to ensure satisfactory performance rates. Here are the configurations which are validated:

```
RAM: 4GB to 192GB
vCPU: 1 to 8 of 2.9GHZ
```

6.3 Installation

Figure 6.1: Connect to your Vsphere Client

In the Virtual Machines tab, in the “File” menu, select “Deploy a new OVF template”.

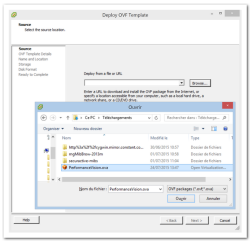


Figure 6.2: Find the Performance Vision OVA file. and Click on “Open”.

The system detects the space available on the disk for the new Virtual Machine, we recommend to allocate the following spaces:

- Trial Virtual Appliance: 4GB RAM, 2 vCPU > 2,0 GHz
- Production:
  - Virtual Poller: 8 GB, 2 vCPU > 2,0 GHz
  - Virtual Appliance: > 16 GB, 4 vCPU > 2,4 GHz

The Virtual Appliance gets installed.

You get notified when the installation is complete.

6.3.1 Get it Started

Once the Virtual Appliance is installed, you have to start it.

6.3.2 Access the virtual console

The probe is launched. When the network interfaces turn into promiscuous mode, click on the Console view and then “Enter” to display the login prompt.

Note: Clicking on the black screen deactivates your mouse. To reactivate it, you can use the key combination `Ctrl + Alt`.

To know how to login and how works the command line interface, please see *Pulsar*. With `Pulsar` you can configure your keyboard, your timezone and the system like IP, DNS, NTP...

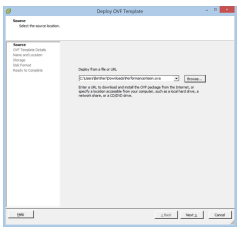


Figure 6.3: Click on “Next”

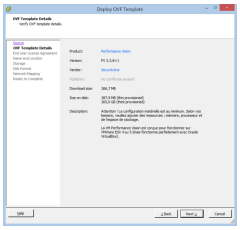


Figure 6.4: Click on “Next”

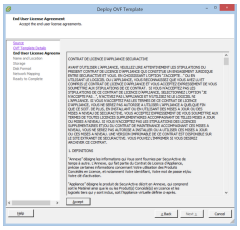


Figure 6.5: Read, then click on “Accept” and “Next”

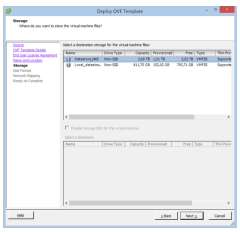


Figure 6.6: Name the Virtual Machine appropriately and click on “Next”.

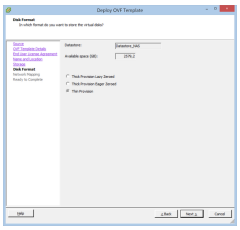


Figure 6.7: Disk configuration

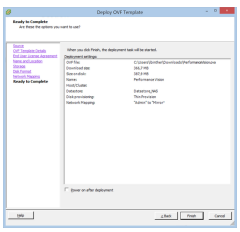


Figure 6.8: Click on “Finish”

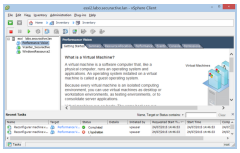


Figure 6.9: Click on “Power on the Virtual Machine” or on the green triangle.

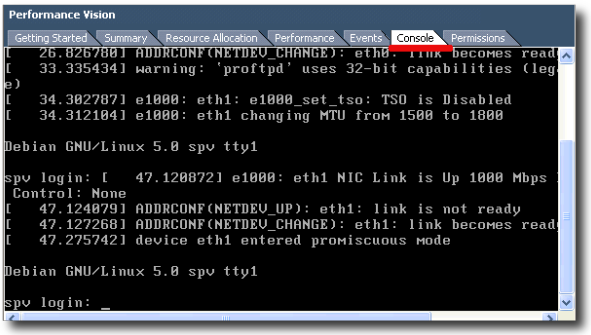

Performance Vision
Getting Started
Summary
Resource Allocation
Performance
Events
Console
Permissions
[26.826780] ADDRCONF(NETDEV_CHANGE): eth0: link becomes read
[33.335434] warning: 'proftpd' uses 32-bit capabilities (leg
e)
[34.302787] e1000: eth1: e1000_set_tso: TSO is Disabled
[34.312104] e1000: eth1 changing MTU from 1500 to 1800
Debian GNU/Linux 5.0 spv tty1
spv login: [47.120872] e1000: eth1 NIC Link is Up 1000 Mbps
Control: None
[47.124079] ADDRCONF(NETDEV_UP): eth1: link is not ready
[47.127268] ADDRCONF(NETDEV_CHANGE): eth1: link becomes read
[47.275742] device eth1 entered promiscuous mode
Debian GNU/Linux 5.0 spv tty1
spv login: _

Figure 6.10: Console login prompt

The summary view provided by Vsphere displays the parameters such as IP addresses:

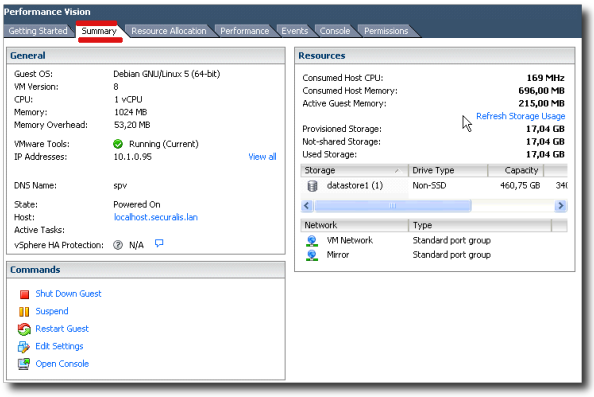


Figure 6.11: Summary View

Note: The virtual machine has a second 150 GB hard disk that you can resize depending on your needs, but then you’d have to format it (via Pulsar’s `format_data_disk` command).

When your probe is setup, you have to reboot the Virtual Appliance.

6.3.3 Insert a license key

Except the empirical virtual appliances of test provided from our Web site, the virtual appliances are delivered without license key. You normally receive this key by e-mail at the product’s delivery. If it is not the case, please contact our sales department: `sales@securactive.net`.

For more information about licensing and how to install the license, please see *Licensing and Upgrades*.

6.3.4 Access the probe interface

To login to the web interface, please see *Access Through a Web browser*.

Then you should check you're license is well configured, to do that see *Licensing and Upgrades*.

6.3.5 Traffic capture

First of all:

- The port mirroring should be activated on yours switches (or TAP eventually)
- Connect the mirror destination port to the ESX server port dedicated to the traffic capture

We will now set the network in Promiscuous mode.

In The following example, we are using an ESX server with 8 physical ports. It is necessary to add a virtual network for traffic monitoring. How to do it?

1. Connect to Vsphere Client

2. Then on your ESX server icon, go to the "Configuration" tab

3. Click on the "Networking" Menu on the left column

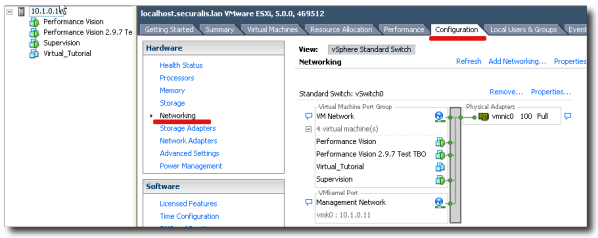


Figure 6.12: Networking Menu

4. Click on "Add Networking"

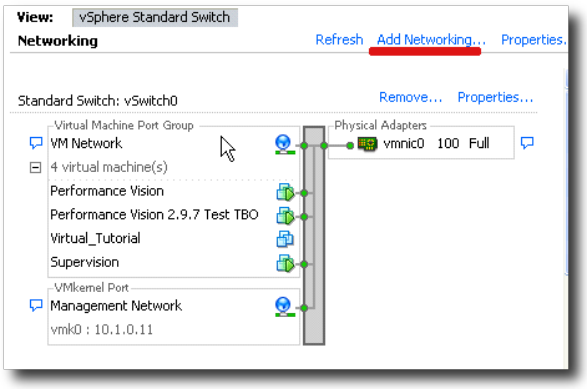


Figure 6.13: Add Networking

Then, on "Network Access" Menu, select the Esx physical port dedicated to the traffic capture (here is vmnic3) and unselect the others. The Esx physical will be bound to the new virtual network (here VM Network2). Click on "Next"

We can customize the new network label as "Mirror" here.

Vlan ID (optional) for vlans tags:

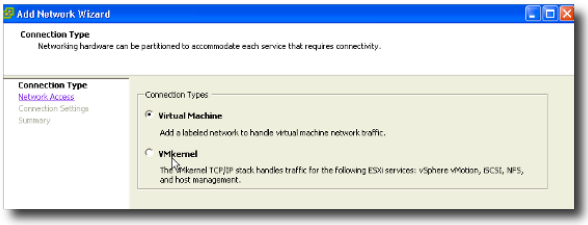

Add Network Wizard
Connection Type
Networking hardware can be partitioned to accommodate each service that requires connectivity.
Connection Type
Network Access
Connection Settings
Summary
Connection Types
Virtual Machine
Add a labeled network to handle virtual machine network traffic.
VMkernel
The VMkernel TCP/IP stack handles traffic for the following ESXi services: vSphere vMotion, iSCSI, NFS, and host management.

Figure 6.14: Select Virtual machine as Connection Types, then Click on “Next”

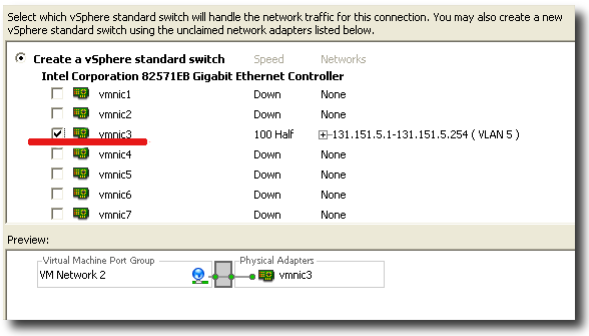

Select which vSphere standard switch will handle the network traffic for this connection. You may also create a new vSphere standard switch using the unclaimed network adapters listed below.
Create a vSphere standard switch
Speed
Networks
Intel Corporation 82571EB Gigabit Ethernet Controller
vmnic1 Down None
vmnic2 Down None
vmnic3 100 Half 131.151.5.1-131.151.5.254 (VLAN 5)
vmnic4 Down None
vmnic5 Down None
vmnic6 Down None
vmnic7 Down None
Preview:
Virtual Machine Port Group
VM Network 2
Physical Adapters
vmnic3

Figure 6.15: vSphere Switch

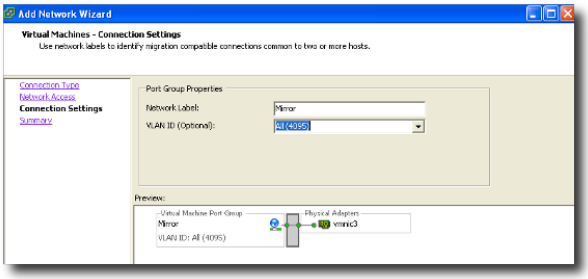


Figure 6.16: The following option allows VLAN tags

```
0    : Disables VLAN tagging on port group
4095: Enables VLAN tagging on port group
```

5. Then click on “Next” and “Finish” to complete the operation.

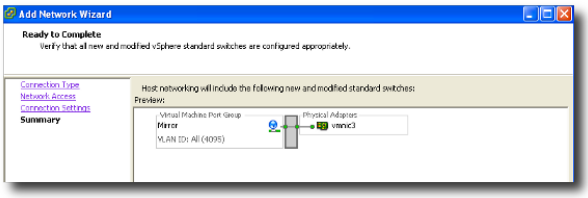


Figure 6.17: Networking Summary

6.3.6 Setup promiscuous parameters.

The Esx Server now manages 2 virtual networks.

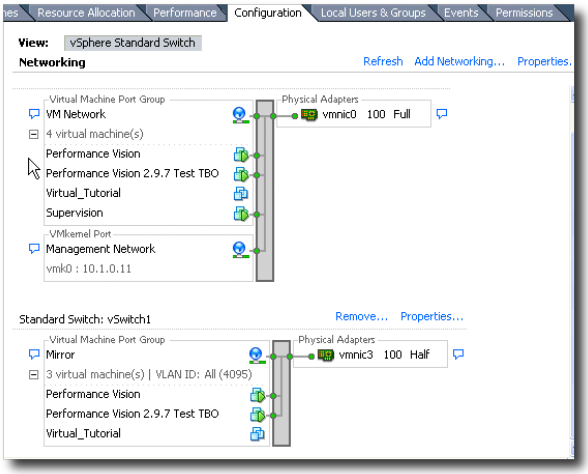


Figure 6.18: Two Virtual Networks

The aim of the second vswitch `vSwitch1` is to show the flows in promiscuous mode.

To set up promiscuous mode on the Mirror Network:

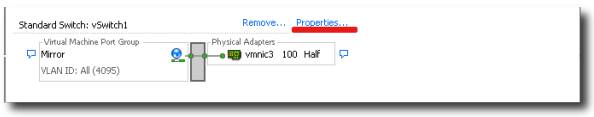


Figure 6.19: Click on «vSwitch1 Properties »

In “General” tab, edit MTU settings to 9000:

The, in “Security” tabs, select “Accept” from the promiscuous mode listbox:

6.3.7 Add a listening network card to virtual appliance.

Here we should add a listening network port in promiscuous mode. Right click of the virtual appliance then choose “Edit settings”.

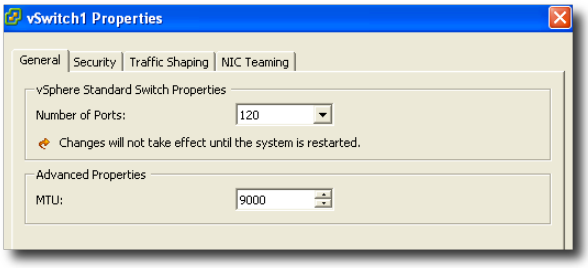


Figure 6.20: General settings: MTU

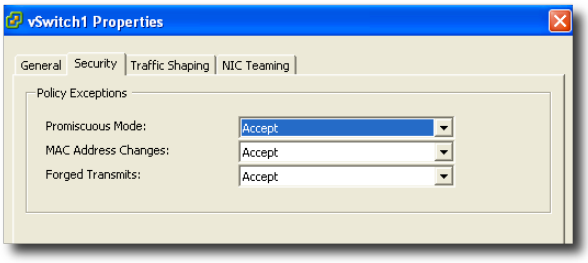


Figure 6.21: Security settings: accept promiscuous mode

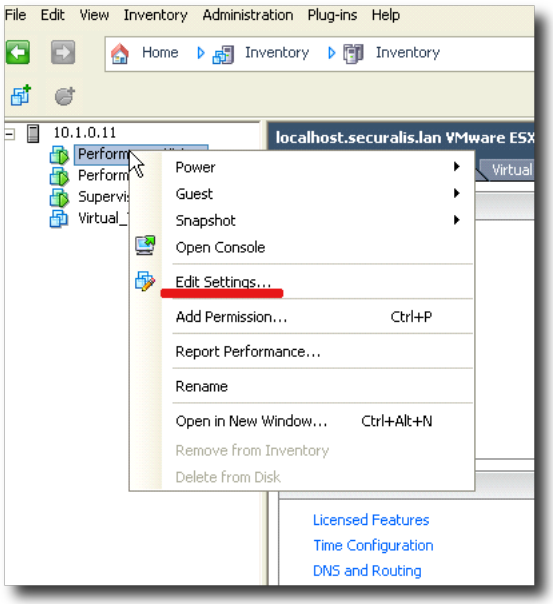


Figure 6.22: Click on Edit Settings

In the Hardware tab, click on “Add”, then choose Ethernet adapter and click on “Next”. Attach the new ethernet adapter to the network in promiscuous mode.

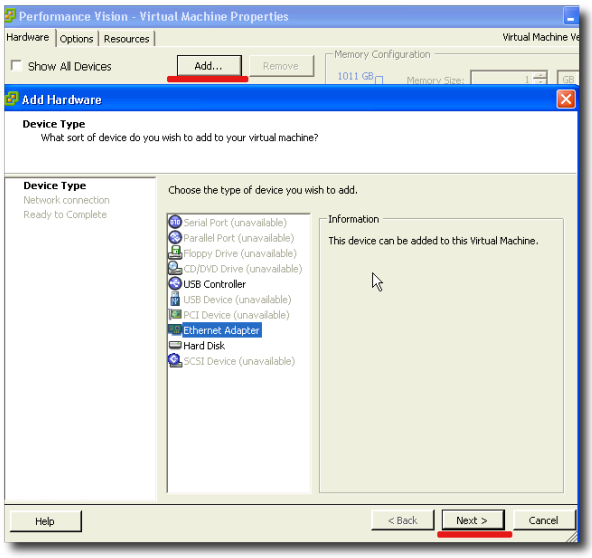


Figure 6.23: Attach Ethernet Adapter

In the network connection listbox, choose the accurate network configured above (Mirror here), then click on “Next”.

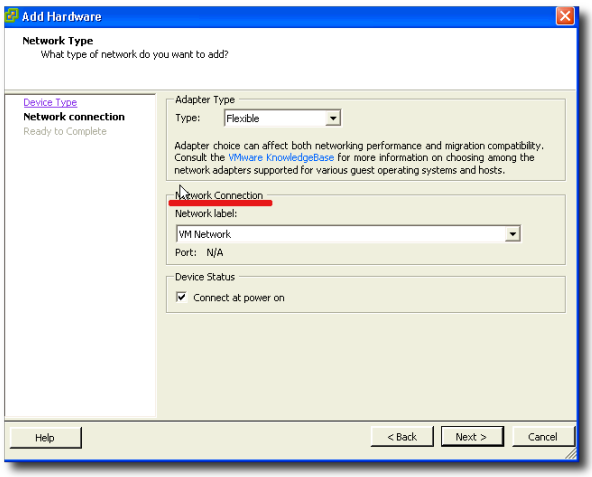


Figure 6.24: Network Connection

Click on “Finish” to complete the operation.

6.4 Validate the traffic capture

You can power on the virtual appliance and validate traffic Capture. There are 2 main methods to validate the traffic capture, with the *graphical interface* (GUI) or with *Pulsar*.

With the **GUI**, as an example, you can monitor the bandwidth after 6 minutes of listening by clicking on the green validation button. See *Use The PV Graphical Interface* for more information about how to use the GUI.

With **Pulsar**, connect via ssh or from the virtual appliance console on the Esx and type `bmon`. See *Pulsar* for more information about the command line interface.

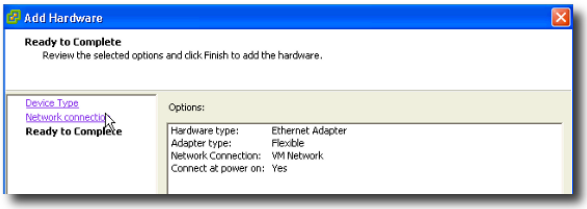


Figure 6.25: Ready to Complete

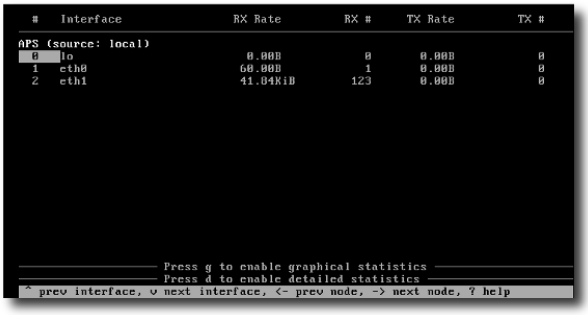

Interface RX Rate RX # TX Rate TX #
APS (source: local)
0 lo 0.00B 0 0.00B 0
1 eth0 60.00B 1 0.00B 0
2 eth1 41.04KiB 123 0.00B 0
Press g to enable graphical statistics
Press d to enable detailed statistics
^ prev interface, v next interface, <- prev mode, -> next mode, ? help

Figure 6.26: Command 'bmon' displays the traffic per interface.

6.5 How to use the product?

The Performance Vision Virtual Appliance is shipped with a default configuration that will likely not match your site very closely. For a better experience it is recommended that you spend some time configuring some additional zones and applications to suit your traffic.

Here are the sections you should consult, in order:

- *User Management* for adding new users;
- *Zone configuration* for adding new zones or modifying the preset configuration;
- *Application configuration* for registering your specific applications;
- *Business Critical Applications* and/or *bcn_config* to define your business critical applications/links;
- *Reports* to schedule periodic reports that will be sent via email.

Eventualy, read the *Use The PV Graphical Interface* then *Interpreting the results* and you will see your network differently.

CHAPTER

SEVEN

CONFIGURATION

7.1 Hardware

The first thing to do is to plug a screen and a keyboard to the probe (for first set-up only) and then to provide electrical power. Once done, just turn power on.

For the screen, the connectivity is a standard VGA port. Two are available, one is located on the front side of the probe, the other is located on the rear side of the probe.

For the keyboard, you can plug it to any of the four USB ports. Two of them are located on the front side of the probe, the two others are located on the rear side of the probe.

By default the probes are equipped with four *Gigabit Ethernet* interfaces labeled `1 to 4`. The first one is the administration port used to connect to the probe. Plug the Gb1 network interface to your network to be able to connect to the probe. The three others interfaces, `2 to 4` are dedicated to network traffic sniffing. Connect one or more of these interfaces to your network according to the network traffic you want to analyze and monitor.

7.2 Pulsar

The probes come with a Command Line Interface named *Pulsar*. This allows the user to check the probe state and configure it when needed.

7.2.1 Connect to the probe

If this is your first encounter with the probe, you will have, for the first time only, to access to the probe physically (just use a screen and keyboard plugged to the probe). Log in with user `admin` and default password `admin`. Once the network address of the the probe will have been set-up you will be able to access to it directly through `SSH` on port `22` also with the same user `admin`.

When logged in you should see the following prompt (version number can vary).

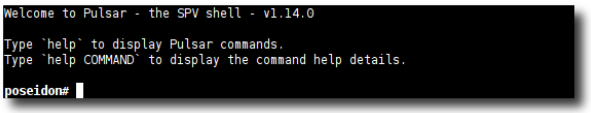

Welcome to Pulsar - the SPV shell - v1.14.0
Type `help` to display Pulsar commands.
Type `help COMMAND` to display the command help details.
poseidon#

Figure 7.1: Pulsar prompt on the *poseidon* probe

Note: Pulsar uses 3 colors while displaying informations.

- **Green** outputs are *informations*.
- **Yellow** outputs are *warnings*.
- **Red** outputs are *errors*.

If needed you can set the keyboard mapping with the `kb <mapping>` command. Typing `kb` displays the list of available mappings.

Pulsar allows you to change the administration password through `passwd` command. This should be your first command. Typing `passwd` in the pulsar shell launches the standard UNIX password-change process.

> **Warning:** At this point, there is no way to retrieve the password. If you totally lost the password, the Securactive support team can generate a new one. See *Support access through VPN*. You can also restore the probe, see *Restore probe state*.

7.2.2 Configure the probe

Use the `config` command to setup up the probe.

```
pulsar# config
service:
1. dns
2. hostname
3. network
4. ntp
5. smtp
6. support
7. **all [default]**
Your choice?
```

Typing `enter` will launch the whole interactive configuration process.

> **Warning:** This command is **mandatory** as it will configure key elements needed for proper operations (DNS servers, hostname, IP address, NTP, SMTP...).

Some changes in configuration require to reboot the probe (command: `reboot`).

7.2.3 Restore probe state

You may need to restore some probe original configuration. There are three way of achieving this. As these are destructive commands a strong confirmation will be requested.

You want to erase any single data from your previous network captures. This preserves configuration settings and IHM user accounts.

```
pulsar# reset data
...
Stopping services...
Deleting data...
Done.
```

The command `reset all` will destroy both your configuration and capture database. You will have a fresh new database. Configuration settings, users and pollers will be reset to default values.

```
pulsar# reset all
...
Stopping all services...
Resetting...
Creating default settings...
Done.
```

7.2.4 Formating data hard drive disks

This is to be used when you are delivered new data disk(s). If you want to use it anyway, any existent data (capture and configuration) will be lost. Default values will be restored.

```
pulsar# format_data_disk

These processses should not be interrupted. Do NOT use Ctrl-C.
Preparing disk ...
Formatting disk ...
Installing disk ...
Generating database ...
This may be quite long (5 min) ...
Done.
```

7.2.5 Listing running processes

The `process` command list all the processes running on the appliance as well as their uptime.

```
pulsar# process
MonitorNevrax
RUNNING: pid 10201, uptime 22:05:18

distribute
RUNNING: pid 10202, uptime 22:05:18

dumptimer
RUNNING: pid 16877, uptime 1:19:02

junkie
RUNNING: pid 16575, uptime 1:19:14

junkie-dumper
RUNNING: pid 16725, uptime 1:19:09

low-space-watchdog
RUNNING: pid 10203, uptime 22:05:18

nevrax
RUNNING: pid 10205, uptime 22:05:18

storage
RUNNING: pid 10204, uptime 22:05:18
```

You can see in this example that some processes have been restarted recently.

Here is the table of all involved processes with a brief explanation:

Name	Run also on a poller	Role
MonitorNevrax	No	Check nevrax resource consumption
distribute	No	Deploy configuration and collect, synchronize and merges CSV files about traffic statistics.
dumptimer	Yes	Signal the end of a 2 minutes statistics collection
junkie	Yes	Network sniffer that computes various statistics about the traffic
junkie-dumper	Yes	Write the statistics into CSV files for the RDBMS
low-space-watchdog	No	Checks available disk space
nevrax	No	Web user interface
storage	No	Stores new data into the RDBMS and handle data aggregation

7.2.6 More about pulsar

`help` provides both global and command help. Tab-completion is enabled for commands and subcommands such as `help`, `config` and `show`.

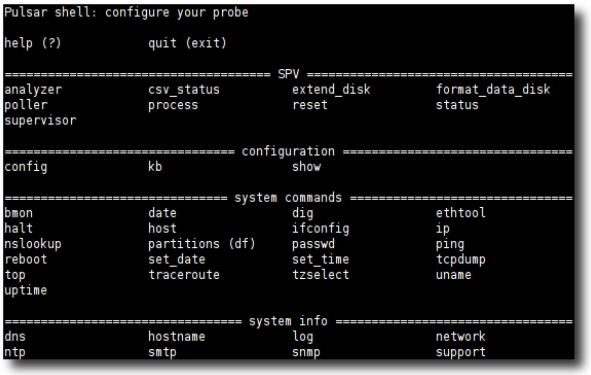


Figure 7.2: Available commands

7.2.7 Configuration example

```
pulsar# config network
[NETWORK]
Connection Type:
1. Static network
2. DHCP
Your choice? 1
IP address: 192.168.1.1
netmask: 255.255.255.0
gateway: 192.168.1.254
```

7.2.8 Support access through VPN

The probes come with an already configured VPN connection to allow access for support operations, if needed. The VPN address is set by default and should normally not be changed. If it needs to be changed, this can be done by the command config and `option 7`. The VPN service is stopped by default. It can be started or stopped at any moment by the corresponding commands `support start` or `support stop`.

Note: In order to have the VPN connection of the probe working fine, you will probably have to configure your network and/or security equipment like your firewalls. Default Host DNS is `vpn.securactive.net` and default port is `443`.

7.2.9 Support with no remote access

In case the probe is not accessible from the Internet, you can use the `diag` command. It'll generate a tarball containing all necessary information for the support team to do the diagnostic. Once the tarball is generated, you'll have to download the file by FTP with the classical admin account (in the /diag directory) and send it to the support team by email or any file sharing platform (file size can be huge).

```
probe# diag
Creating a diagnostic package, could be long...
Download the diagnostic file with a FTP client: DIAG-E2A346A2-55F7-5834-40AE-B1EC5967FB61-2014061
```

7.2.10 What information to give to the support team

Use the `info` command: This command summarizes all the basic information needed for support assistance, such as state of the probe, ip address of support tunnel and other useful information.

```
probe# info
UUID                        E2A346A2-55F7-5834-40AE-B1EC5967FB61
Platform                    vmware
Role                        Collector
Release                     3.0.9-r1-internal
License                     Valid license (2015-01-02 00:00:00)
/ usage                     42%
Datadisk present?           True
/srv usage                  27%
Admin interface             192.168.80.119 / 255.255.255.0 - 192.168.80.255
Support interface           10.10.0.5 / 255.255.255.0
Sniffer state               RUNNING
Distribute state            RUNNING
Storage state               RUNNING
IHM state                   RUNNING
Database up?                True
```

7.2.11 How to configure User Interface language ?

User interface is available in English and French languages. The language is detected automatically based on the default language of the browser used to access the probe. So, to get the User interface to use the desired language, the administrator should check and configure the default language of its browser.

7.3 SPV Functional Configuration

7.3.1 User Management

There are two groups of users in the "Users Configuration" interface:

- The *Administrators* group
- The *Users* group

These two groups have different access permissions to the application pages: the administrators group provides its members a full access to the "Configuration" pages. Users group members will be able to read reports but will not have access to the configuration page.

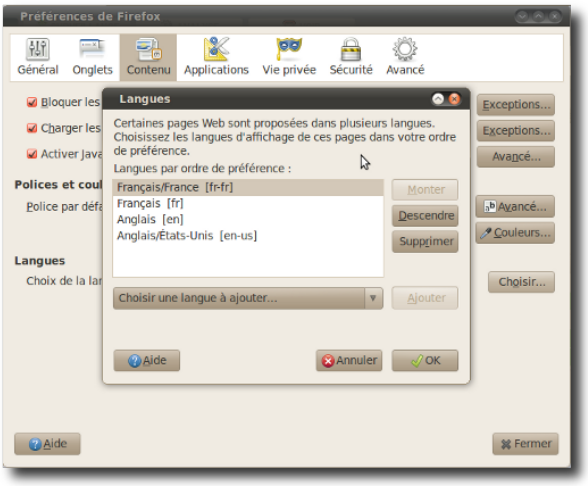


Figure 7.3: Configuration of the French language in Firefox

In order to create a new user account you must be logged into the appliance as a member of the *Administrator* group. As mentioned in the above paragraph, the default admin group has the right to create, modify and access the configuration. You can add a new user account by clicking on the `Users` tab found on the configuration menu on the left hand side. Then click on the `Add` button and fill in the "User information" (username, password, and group). Make sure the `Active` button is checked, otherwise the user won't be able to login. Thanks to this option you will be able to disable or enable an account without deleting it.

Example: Adding a new member to Administrators group. In the example below, we have created a user account in the `Administrators` group with the user name `John` and `foo2` as the password:

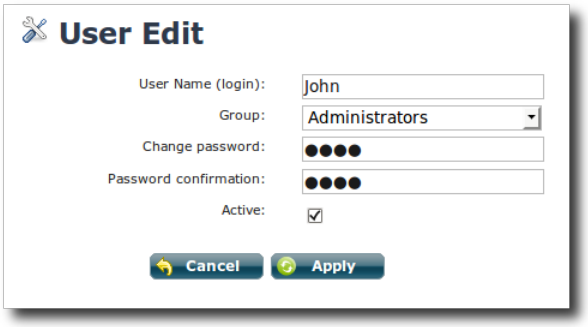


Figure 7.4: Edit User

The user name is case sensitive, and it is required to be non-empty and to contain only letters, numbers, or _ (underscores).

You can modify a user account by clicking on the `Users`" tab found on the configuration menu on the left hand side, and then clicking on the user name of the desired user account in the user list. You will be able to modify any field on a created user. Please note that the password field will appear empty on edition to avoid giving out information and will not be modified upon edition if it is left empty. In order to save any modifications, click on the `Apply` button.

You can delete a user account by clicking on the "Users" tab found in the configuration menu on the left hand side, and then clicking on the check box next to the user name of the account you wish to delete. Then, clicking on "Delete" button will delete all selected Users.

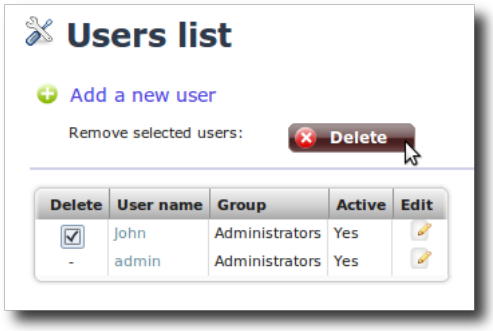


Figure 7.5: User account 'John' is about to be deleted.

7.3.2 Zone configuration

The aim of this chapter is to help the administrator of the platform to configure zones. When you change or create a *zone*, the modifications will be effective within a short delay for future integrated data but not to the already captured data which keep their old zone attribute.

How to access the configuration menu?

After clicking on the top right `configuration` button, you will observe a tree configuration menu with different items.

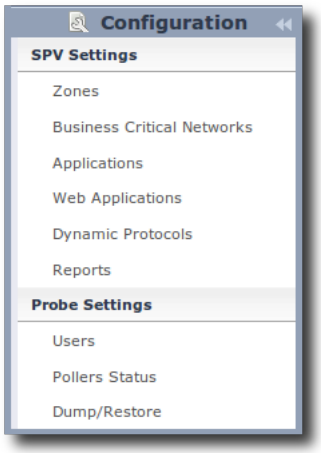


Figure 7.6: Configuration menu

Zones management using the GUI

Pleaser refer to *Zones* for *Zone* tree and *Fallback* explanations.

You can reach the zone configuration page by clicking on the `Zones` label of the menu. The illustration below lists the zones and their corresponding definitions.

This page allows you to add a zone, edit a zone, move some zones around or delete a zone.

In order to edit a zone or add a new child you need to click on the zone block to expand it. All its filters will then be available for edition.

Each filter is composed of a subnet field, a MAC address field, a Vlan field, and device and poller select boxes. Any of these filters can be left blank, in which case no tests are performed on the corresponding value. When

Fallback 1 rules.
Internet 1 rules.
IPv6 RFC 1 rules.
Routable Unicast RFC 3587 1 rules.
6to4 RFC 3056 1 rules.
Doc RFC 3849 1 rules.
Teredo RFC 4380 1 rules.
Routable Unicast RFC 3587_fallback (Zone fallback) 1 rules. -10
IPv4 into v6 1 rules. -20
IPv4 1 rules. -100
IPv6 1 rules. -100
Private 1 rules.
Broadcast 2 rules.
Local 1 rules.
APIPA RFC3927 3 rules.
Clients 1 rules.
LAN 1 rules.
Building 1 1 rules.
Building 2 1 rules.
Financial 1 rules.
IT 1 rules.
LAN Guests 1 rules.
LAN gwa 1 rules.
Management 1 rules.
Sales 1 rules.

Figure 7.7: Overview of the zone tree editor

several fields are set, all must match simultaneously for a conversation to be associated with this zone (in other word, the filters are logically anded together).

Here are some examples of valid subnets:

- `192.168.100.0/24`
- `192.168.100.12/32`
- `::ffff:192.168.0.0/128`

and valid MAC addresses:

- `32:43:a0:00:00:01`
- `32:43:a0:00:00:01/20`

Finally, the numeric priority field allows to alter the default priority (`0`), greater priorities being tested before lower priorities. Note that priorities can be negative values as well as positive values.

Zones management using an external file

Alternatively, one can export the zone configuration onto a `CSV` [1] file that can be edited using any spreadsheet program, and imported again.

7.3.3 Application configuration

You can configure *Applications* in the configuration page. Applications represent the business applications running on your network and make the reports provided easily understandable to everyone in your organization.

To access the configuration of *Applications*, click on the `Configuration` button, on the top right of the user interface.

To create an application, go the to `Application` submenu, in the left menu.

This panel displays the existing Applications (by default or user defined). To create an Application, click on `Create new application`; you will see the configuration screen.

An Application can be defined using the following elements:

[1] Comma Separated Values

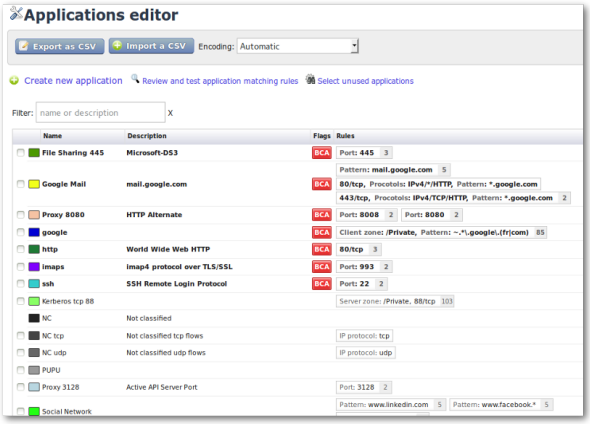


Figure 7.8: Application list screen

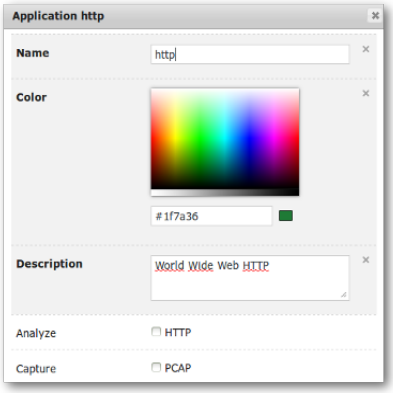


Figure 7.9: Application configuration screen

- **Name**: it corresponds to the designation of each Application, which will be used in displays. This is a mandatory field.
- **Color**: it is the color which will be used to display this specific Application in graphs. This is a mandatory field.
- **Description**: it is a description field, which should be used to track information related to this Application.
- **BCA/HTTP/PCAP** flags: to mark this application as **Business Critical**, as requiring HTTP analysis and/or automatic traffic capture.

Then, using the `Add rule` button, you can attach as many rules to this application. A flow that will match any of these rules will be associated with this application. A rule can test any combination of:

- *IP Protocol*: to select a given IP protocol (such as 'TCP' or 'UDP').
- *Application Port Range*: single port or port range.
- *Client and Server IP Address*: IP addresses (or ranges) of the clients and servers of this application.
- *Client and Server Zone*: zone in which the clients and servers are located (see *Types of Conversations* for details on client/server identification).
- *Protocol Stack*: allow to select only those flows identified by the sniffer as featuring this protocol stack (use with caution).
- *Poller*: to filter on a single poller.
- *Device Identifier*: to filter on a single network adapter.
- *VLAN*: to select flows from a given Ethernet VLAN tag or range.
- *Ethernet Protocol*: to filter on a given Ethernet protocol.
- *Client and Server Ethernet Address*: MAC addresses (or ranges) of the clients and servers of this application.
- *Web Application Pattern*: allow to select those HTTP messages concerning only a given URL pattern.

Web Application Pattern

The web application pattern in an application rule is used to identify specifically HTTP applications. They are defined as patterns matched against the URLs contained in HTTP requests. The patterns should contain at least a domain name, optionally including wild card characters like '*' or if you check 'regex mode', you can set POSIX regular expressions.

Notice that in a typical conversation, several HTTP messages referring to several URLs will be present. The application rule will only be checked against the first encountered URL per socket. This is not a problem if all URLs in a given socket follow the same pattern (which is usually the case provided your pattern is not too picky).

7.3.4 Business Critical Applications

Any application can be tagged as **Business Critical**. Those applications are used to display the *'Business Critical Application Dashboard'*. To flag a given application as critical (or remove this flag), edit this application by clicking on it on the application list and toggle the *Critical Application* checkbox.

When you flag an application as *Critical*, three additional parameters are requested:

- **The minimum transaction count**. It indicates, for one minute, the minimum of SRT (Server Response Time) events to be seen on the network for being considered as a pertinent measurement. If no transaction at all is seen during the period of time analyzed, the color displayed on the *BCA dashboard* will be "white". If the number of events seen during the period of time analyzed is above zero but under this value, the color displayed on the BCA dashboard will be "grey". It means that some events have been seen, but not enough to be considered as a pertinent measurement. If the number of events seen during the period of time analyzed is above or equal to this value, the color displayed on the *BCA dashboard* will be either "green", "orange" or "red" depending on the *EURT* values.

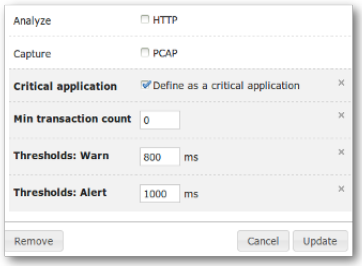


Figure 7.10: Business critical application edition

- **The warning threshold level of the EURT (End User Response Time) value in milliseconds**. When the value is above or equal to this level, the color displayed on the BCA dashboard will be "orange". When the value is under this level, the color displayed on the *BCA dashboard* will be "green".
- **The alert threshold level of the EURT value in milliseconds**. When the value is above or equal to this level, the color displayed on the *BCA dashboard* will be "red".

Note: To be useful and pertinent, these parameters must be accurate values adjusted to your network configuration. These values can be easily changed for fine tuning or to cope with any change in the network or applications you are using.

A new critical application will benefit of all the data history: after having defined an application as critical, if the data has already been collected for this application then the thresholds levels will be automatically applied on the *BCA dashboard*, even for a period back in time.

7.3.5 Business Critical Networks

A BCN consists of a *virtual link* between two zones; its objective is to monitor normal volume and performance levels between two network segments, which represent a strategic network link for your organization (e.g. link from the data center to a remote site, from the server VLAN to a user VLAN). An administrator can configure thresholds for `warning` and `alert` on bandwidth consumption, Retransmission Rate (RR) and Round Trip Times (RTT).

A specific configuration screen allows configuring the specified BCN. To access it, just go to the *Configuration* menu and choose the entry labeled *Business Critical Networks*.

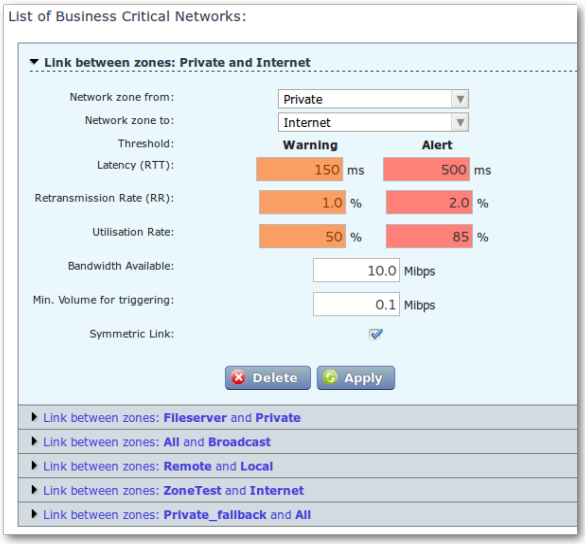


Figure 7.11: Editing an existing Business Critical Network

From here you can add a new BCN or edit the parameters of an existing BCN. Modifications will also be applied on already captured traffic.

For each Critical Network, you have to configure the following parameters:

- The source/destination network zones.
- One or several thresholds for both `Warning` and `Alert` levels, all these thresholds are computed from source to destination and not from client to server. We call this an "oriented" metric:
  - Oriented latency (RTT in ms)
  - Oriented retransmission rate `(%)`
  - Utilization rate `(%)` according to bandwidth available `(Mib/s)`
- A minimum volume for triggering `(Mib/s)`. This value represents the minimum bandwidth observed from which you will consider the performance and volume thresholds as relevant.
- The thresholds values can be configured as symmetric by ticking the `Symmetric Link` check-box or be configured as distinct values for both directions. This is particularly useful when the critical network:
  - refers to **asymmetric** connections like ADSL,
  - has one of its zones **closer** to the poller than the other zone and latency `(RTT)` computation is impacted (see *Distributed Architecture*).

You can define thresholds from either one criterion or more (any of the following: latency, retransmission rate and consumption level). But you cannot define a BCN from one zone to itself, as their intended purpose is to check the performance of most important links or routes between two network segments.

By applying your changes, the *BCN Dashboard* will be updated in accordance with the new threshold values (including already captured data). To be useful and pertinent, these parameters must be accurate values adjusted to your network configuration. These values can be easily changed for fine tuning or to cope with any change in the network or applications you are using.

7.3.6 Reports

Creating *Reports* is just a matter of a few clicks. You can easily create and define exactly the level of information you want to get. You will receive it directly in your mailboxor via *FTP* at the frequency you prefer.

Configuration

In the first step, you start by creating a template that will mainly define the name of the report, the list of recipients, a description and the scheduling settings. In the second step you just have to add the different views you want to see to the appropriate template. Then you're done, just check your mailbox.

To create a report template, in the *Configuration* area, select `Reports` in the menu list on the left. This will display the list of existing report templates. Use the button `Create` to create a new report template. Please note that this feature is only available for users with administration rights.

To create a report template you must fill some information:

- The name of the report for easy identification purpose,
- The full description of this report which will be copied in the PDF file generated.
- The language option defines the language that will be used for this reports (thus the language for the report can be different than the language of the web screen),
- The list of recipients defines the email addresses to which the reports will be sent (the recipients email addresses can be separated by a comma, a semi-colon or a new line),
- Scheduling settings define the frequency at which the reports will be sent.Available options are:
  - `Day`: Generates the report every *x* day(s); example: every two days.

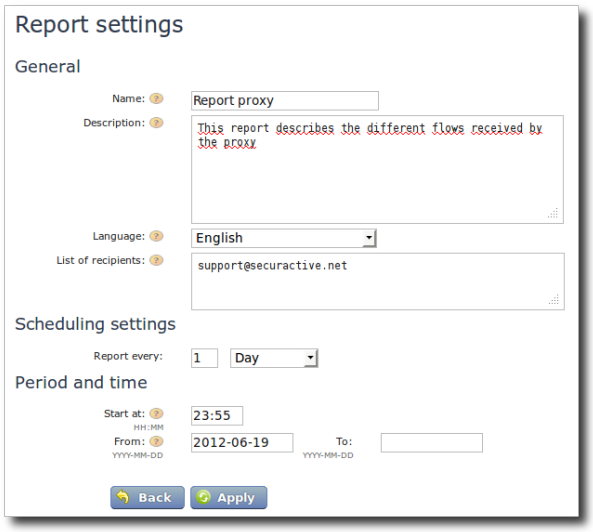


Figure 7.12: Create a new report

  - `Week`: Generates the report every *x* week(s) the selected days; example: every two weeks on Friday (several days in the week can be chosen).
  - `Month`: Generates the report every *x* month(s) on *y* day; example: every month, the first of the month (be careful, if you choose the day `29`, `30` or `31`, you will only receive your reports if there is such day in the corresponding month).

- `Start at` defines the hour (format `HH:MM`) at which the generation of the report will start. Once the report will have been generated it will then be sent to the recipients email addresses.
- `From` and `To` fields are optional. This allows you to define a validity period for the report. In such case, the report will only be sent in the period ranging from the first date up to the second date.

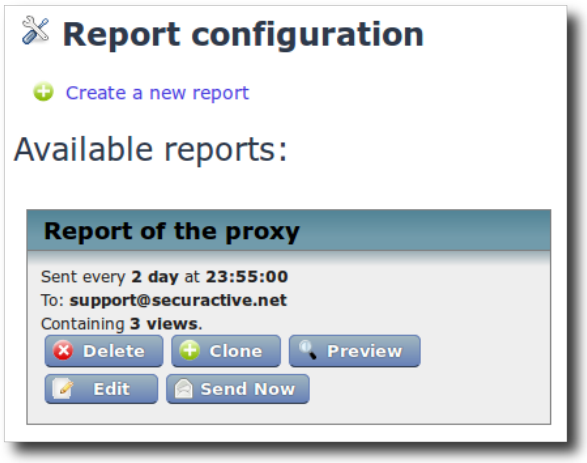


Figure 7.13: Report: A template just created

The new report template just created will appear in the list of available report templates. A summary is displayed (scheduling frequency, generation time, first recipient emails). At this stage it is empty and does not contain any view, this is why you have `Containing 0 views` indicated. After having added some views to the report, here will be indicated the number of views contained in the report.

Add views to report

To add a view to a report template, just go to the screen with the desired view. Select a time period and run the search. Once search is completed, the link `Add this page to a report` becomes active. When you click on it, a drop box with the list of available template reports is displayed. You can chose the template report to which you want to add the current view and click on the button `Add`. If you need, you can click on `Show report list`, it will open the configuration area with the list of available report templates.

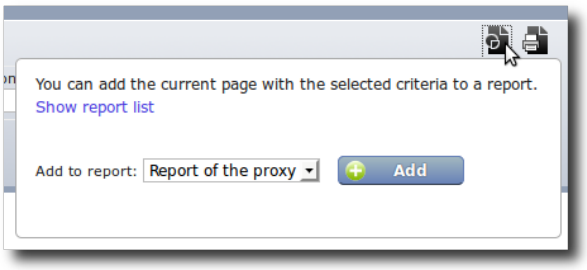


Figure 7.14: Add a view to a report template

Please note that while the time is fixed, the date will remain relative to the moment the report is sent. If the view you're adding starts yesterday at `20:00` and ends today at `8:00`, and the report is scheduled to be sent next Friday, then the effective capture time bracket will be from Thursday at `20:00` to Friday at `8:00`.

Once the page is added to a report, you can modify the filters with the `Edit` button. For each page, you may add an optional description to explain its purpose.

Note: Before release 2.9, an additional time delta was added under certain circumstances. As of 2.9 it's not longer the case; all dates are relative to the day the report is being sent.)

Actions on reports

A report template can be deleted with the button `Delete`. You can clone a report template: all its parameters and included views will be duplicated. A new report template is created with `(copy)` added to the report name.

`Preview` will start the generation of the report right now and you will be able to see the PDF file with your favorite PDF viewer once it has been generated. Notice that the generated report will query the same time intervals than the next scheduled report. This can lead to some blank pages if the data for these intervals were not collected yet.

`Edit` allows you to change the parameters of the report template (name of the report, the list of recipients and the scheduling settings...).

`Send now` will start the generation of the report right now and the report will be sent by mail once it has been generated. Again, the report will query the same time interval than the next scheduled report.

Sending Email

So that your reports could be sent properly to the recipients email addresses, you need to configure the *SMTP* server within *Pulsar*. You can do that with the `config smtp` command. Then just add a valid *SMTP host*, and in option a *login* and *password* if you use an authenticated *SMTP* server. You also can modify (with the same command) the `From` header of the emails generated by the probe.

After that you can either reboot the probe or use `smtp stop` followed by `smtp start` commands to activate the new configuration.

7.3.7 Degradation

This configuration page allows to change how the aggregator system will merge the data. This merging is done per metric. You can also tell to the aggregator system to not degrade a metric at all. For each item there is an embeded help, here some additional information:

IP degradation is done in two passes, the first one is *zoned*, you are supposed to set your `Internet` zone or equivalent. Then a second step of IP degradation is available for all IPs.

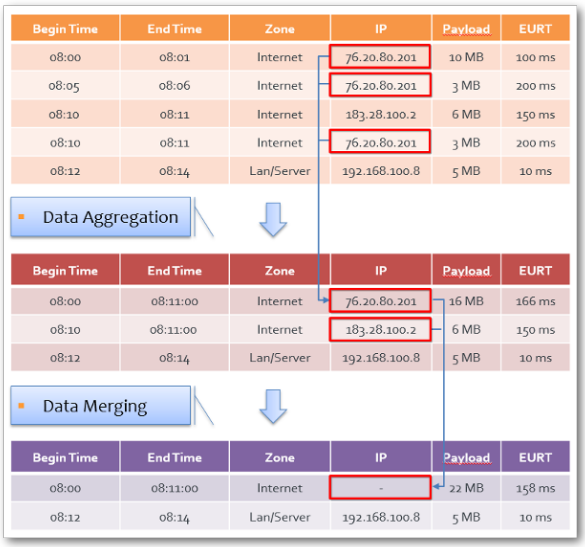


Figure 7.15: IP aggregation, then degradation

7.3.8 SNMP

Optionally, *SNMP* requests are answered on default `SNMP port` (see *Pulsar* documentation). The *SNMP* objects that are thus made available are twofold. First there are the standard *SNMP* objects then SPV specific objects. Version *2c* of the protocol is supported.

System MIB

The probe uses the UNIX `Net-SNMP` [2] daemon, which serves standard *MIB*. So you can monitor your probe from your SNMP console as you would normally monitor any UNIX server. For instance the usual statistics about network interface usage, file system available spaces, I/O operations, etc, are available.

Monitoring specific MIB

In addition to these default information the probe provides various statistics under `iso.org.dod.internet.private.enterprises.securactive`.

The comprehensive *MIB* files are available from our web site [3] so this section only sketches what kind of information is available. You are encouraged to download the actual *MIB* for use with your common purpose *SNMP* console. This will give you access to:

- *Interface statistics* for each network interface, such as the count of received packets, dropped packets and duplicated packets.

[2] http://www.net-snmp.org
[3] http://www.securactive.net/en/documents/250-securactive-mibs/download

- *Protocol statistics* for each recognized protocol, which can give a good impression on the realtime composition of the whole network stream.
- *Various CPU/RAM information* that are destined to troubleshoot an SPV more than to reveal anything about the network.
- *License related information* such as date of expiry and so on.
- *Averaged metrics* such as *RTT* or *DNS* response time.

BCN and BCA MIBS

Since the 2.9 version, two new modules are available: BCA and BCN. Please update your MIB file if you use a SPV MIB before 2.9. Here is a tree description of the BCA and BCN MIB:

BCA module

```
+--sactSPVBCAModule(1)
   |
   +--spvBCAStateTable(1)
   |  |
   |  +--spvBCAStateEntry(1)
   |     |  Index: spvBCAName
   |     |
   |     +-- -R-- String    spvBCAName(1)
   |     +-- -R-- EnumVal   spvBCAStatus(2)
   |     |        Values: Ok(1), Warning(2), Alert(3), NA(4), Nothing(5), NotEnough(6)
   |     +-- -R-- Gauge     spvBCAEURT(3)
   |     +-- -R-- Gauge     spvBCASRT(4)
   |     +-- -R-- Gauge     spvBCASRTCount(5)
   |     +-- -R-- Counter   spvBCASRTCountSum(6)
   |     +-- -R-- Gauge     spvBCARTTClient(7)
   |     +-- -R-- Gauge     spvBCARTTServer(8)
   |     +-- -R-- Gauge     spvBCADTTClient(9)
   |     +-- -R-- Gauge     spvBCADTTServer(10)
   |     +-- -R-- Gauge     spvBCATrafficClient(11)
   |     +-- -R-- Gauge     spvBCATrafficServer(12)
   |     +-- -R-- Counter   spvBCATrafficClientSum(13)
   |     +-- -R-- Counter   spvBCATrafficServerSum(14)
   |     +-- -R-- Gauge     spvBCAThresholdMinSRTcount(15)
   |     +-- -R-- Gauge     spvBCAThresholdWarning(16)
   |     +-- -R-- Gauge     spvBCAThresholdAlert(17)
   |
   +--spvNevraxBCATime(2)
```

BCN module

```
+--sactSPVBCNModule(2)
   |
   +--spvBCNStateTable(1)
   |  |
   |  +--spvBCNStateEntry(1)
   |     |  Index: spvBCNName
   |     |
   |     +-- -R-- String    spvBCNName(1)
   |     +-- -R-- String    spvBCNZoneA(2)
   |     +-- -R-- String    spvBCNZoneB(3)
   |     +-- -R-- EnumVal   spvBCNGlobalStatus(4)
   |     |        Values: Ok(1), Warning(2), Alert(3), NA(4), Nothing(5), NotEnough(6)
   |     +-- -R-- EnumVal   spvBCNStatusAtoB(5)
```

```
|     |        Values: Ok(1), Warning(2), Alert(3), NA(4), Nothing(5), NotEnough(6)
|     +-- -R-- EnumVal   spvBCNStatusBtoA(6)
|     |        Values: Ok(1), Warning(2), Alert(3), NA(4), Nothing(5), NotEnough(6)
|     +-- -R-- Gauge     spvBCNRttAtoB(7)
|     +-- -R-- Gauge     spvBCNRttBtoA(8)
|     +-- -R-- Gauge     spvBCNRrAtoB(9)
|     +-- -R-- Gauge     spvBCNRrBtoA(10)
|     +-- -R-- Counter   spvBCNRetransCountSumAtoB(11)
|     +-- -R-- Counter   spvBCNRetransCountSumBtoA(12)
|     +-- -R-- Gauge     spvBCNBandwidthAtoB(13)
|     +-- -R-- Gauge     spvBCNBandwidthBtoA(14)
|     +-- -R-- Counter   spvBCNTrafficSumAtoB(15)
|     +-- -R-- Counter   spvBCNTrafficSumBtoA(16)
|     +-- -R-- Counter   spvBCNPacketsCountSumAtoB(17)
|     +-- -R-- Counter   spvBCNPacketsCountSumBtoA(18)
|     +-- -R-- EnumVal   spvBCNThresholdSymetricLink(19)
|     |        Values: True(1), False(2)
|     +-- -R-- Gauge     spvBCNThresholdBandwAvailableAtoB(20)
|     +-- -R-- Gauge     spvBCNThresholdBandwAvailableBtoA(21)
|     +-- -R-- Gauge     spvBCNThresholdBandwMinAtoB(22)
|     +-- -R-- Gauge     spvBCNThresholdBandwMinBtoA(23)
|     +-- -R-- Gauge     spvBCNThresholdBandwrateWarningAtoB(24)
|     +-- -R-- Gauge     spvBCNThresholdBandwrateWarningBtoA(25)
|     +-- -R-- Gauge     spvBCNThresholdBandwrateAlertAtoB(26)
|     +-- -R-- Gauge     spvBCNThresholdBandwrateAlertBtoA(27)
|     +-- -R-- Gauge     spvBCNThresholdRttWarningAtoB(28)
|     +-- -R-- Gauge     spvBCNThresholdRttWarningBtoA(29)
|     +-- -R-- Gauge     spvBCNThresholdRttAlertAtoB(30)
|     +-- -R-- Gauge     spvBCNThresholdRttAlertBtoA(31)
|     +-- -R-- Gauge     spvBCNThresholdRrWarningAtoB(32)
|     +-- -R-- Gauge     spvBCNThresholdRrWarningBtoA(33)
|     +-- -R-- Gauge     spvBCNThresholdRrAlertAtoB(34)
|     +-- -R-- Gauge     spvBCNThresholdRrAlertBtoA(35)
|
+--spvNevraxBCNTime(2)
```

Note: Notice that none of these *MIB* objects is currently settable.

7.3.9 TLS Decryption

Some of the protocols inspected by SPV may be encrypted using TLS, namely: HTTP, SKINNY, SIP. Under some conditions SPV can decrypt these streams and proceed with inspection as normal. In other words it is possible to visualize HTTPS transactions.

To activate this feature you must fulfill all of the following requirements:

- Have access to the private keys of the targeted servers and upload them into SPV. Please bear in mind that anyone with your private key can do the same as SPV probe, so make sure you upload it using HTTPS and secure access to the probe file system.
- Force the server (or the client) to use these keys to encrypt the handshake (in other words, disable those encryption algorithms such as Diffie-Hellman). For instance, for the Apache web server, make use of the *SSLCipherSuite* parameter. Here for instance we allow only the cipher suite using RSA key exchange algorithm:

  ```
  SSLCipherSuite kRSA
  ```

- Force the servers or clients to forget about previous TLS sessions (or wait long enough, typically some hours). SPV will make its best to remember new TLS sessions but will dedicate only a limited amount of memory to do so. Also, memorized sessions are not written to disk and so will not survive restarting the sniffer.

- Make sure the probe will receive 100% of the traffic to/from targeted servers as decryption can not work around missing packets.
- Make sure required resources are available since decryption is CPU intensive.

CHAPTER

EIGHT

INTERPRETING THE RESULTS

Note: Note about terms used: starting from version 2.8, The in/out notion has been fully replaced by Server/Client. So in our Graphs, any RTT and RR (in/out) should be considered as RTT,RR (Server/Client) as in the following rules.

- RTT in stands for RTT Server.
- RTT out stands for RTT Client.
- RR in stands for RR Server.
- RR out stands for RR Client.

8.1 Business Critical Application Dashboard

To customize this view for your own needs, just go to the *Configuration* menu and choose the application you want to be a 'business' one. (see the *Business Critical Applications*).

The purpose of the Business Critical Application Dashboard (BCA) is to have, regrouped into one single view, the most important elements that are critical for your business. In one single screen vital information is presented to people in charge in order to radically improve early diagnostics and impact analysis. The right information is directly available through a completely configurable and dynamic dashboard view. What is monitored is the EURT (End User Response Time) metric. Thus, this dashboard reflects the quality of experience of the users for the selected critical applications.

- In red: poor quality
- In orange: medium quality
- In green: good quality
- In grey: not enough data gathered

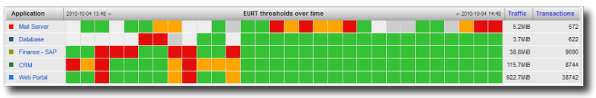


Figure 8.1: Business Critical Application Dashboard view

8.1.1 Business Critical Application Dashboard Capabilities

- You can customize the business critical dashboard to view specific applications and metrics corresponding to your specific business.
- From the BCA dashboard, you can drill-down from the general view to detailed analysis and problem resolution views.

Figure 8.2: Quick links in the Business Critical Application Dashboard view.

Thus, from each Business Critical Application, with a single click on the appropriate icon, you can:

- Directly access to the corresponding Application Dashboard,
- Add a filter on this specific Critical Application (in case you have defined a lot of Critical Applications and you want to see only one for a moment),
- Edit Application characteristics.
- Directly access to the details of Conversations for this Application.

Note: If you click on the icons that are next to the name of the application at the beginning of each line, the quick links will take into account the complete period of time currently displayed. If you click on the icons associated to a specific period of time, the quick links will used this specific period time when redirecting you to a detailed screen.

- You will always see up-to-date information with the auto-refresh feature of the BCA dashboard. The information will be automatically refreshed based on the data aggregation level (see *aggregation period*). For example if the "Aggregate level" is "2 minutes", the BCA will be updated every two minutes; if the "Aggregate level" is "15 minutes", the BCA will be updated every fifteen minutes.

8.2 Business Critical Networks Dashboard

To customize this view for your own needs, just go to the *Configuration* menu and choose the entry labeled `Business Critical Network` (see the *Business Critical Applications*).

The Business Critical Network Dashboard (BCN) is aimed at presenting in a single screen the status of your organization's most critical network "links". You can customize the business critical network dashboard to view the status of the most strategic links corresponding to your business.

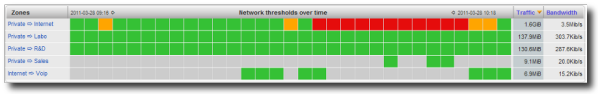


Figure 8.3: Business Critical Network Dashboard

From the Business Critical Network Dashboard, you can drill down from the general view to more detailed information for analysis and problem resolution:

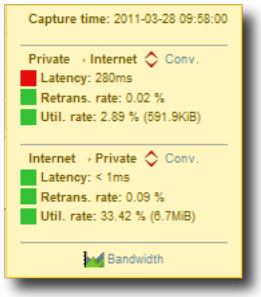


Figure 8.4: Detailed values for a point of time

By pointing with the mouse, you can view the threshold values for each direction at each point of time (indicating status OK, Warning or Alert as well as the value for each direction). You can also access to the bandwidth graphs and the conversations table for each link. If you click on the icons that are next to the name of the link at the beginning of each line, the quick links will take into account the complete period of time currently displayed. If you click on the icons associated to a specific period of time, the quick links will used this specific period time when redirecting you to a detailed screen. You will always see up-to-date information with the auto-refresh feature of the *BCN dashboard*. The information will be automatically refreshed based on the data aggregation level (see *aggregation period*). For example if the "Aggregate level" is "2 minutes", the BCN will be updated every two minutes; if the "Aggregate level" is "15 minutes", the BCN will be updated every fifteen minutes.

8.3 VoIP Module

A specific reporting for *Voice over IP* traffic is provided. The aim of this module is to show the volume and quality of service associated with VoIP flows.

8.3.1 Supported protocols

These VoIP protocols are supported:

- SIP + RTCP + RTP
- MGCP + RTCP + RTP
- SKINNY + RTCP + RTP

For more information, please consult the corresponding *RFCs*:

- SIP as defined in **RFC 3261** (http://tools.ietf.org/html/rfc3261.html)
- MGCP as defined in **RFC 3435** (http://tools.ietf.org/html/rfc3435.html)
- RTP as defined in to **RFC 3550** (http://tools.ietf.org/html/rfc3550.html) and **RFC 3551** (http://tools.ietf.org/html/rfc3551.html)
- RTCP as defined in **RFC 3605** (http://tools.ietf.org/html/rfc3605.html)

8.3.2 Basics of VoIP

Voice Over IP relies on three protocols to operate over IP networks:

- **Signalization protocol**: the role of this protocol is to establish and control the voice communications. It usually consists of communications between the IP phone and a call manager / IPBX. The 2 signalization protocols supported are SIP (Session Initiation Protocol) and MGCP (Media Gateway Control Protocol). Please note that SIP may follow the same route as the RTP traffic or not, while MGCP follows the same route as RTP.
- **Media protocol**: the role of this protocol is to carry the voice signal from one IP phone to the other IP phone (it can eventually go through the call manager / IPBX). RTP is the only media protocol supported by *Performance Vision*. It stands for *Real Time Protocol*; it usually runs over UDP.
- **Control protocol**: the role of this protocol is to carry quality and control information from one phone to the other phone. RTCP is the only control protocol supported. It stands for *Real Time Control Protocol*.

8.3.3 Quality of service & MOS

MOS stands for Mean Opinion Score. It is a numeric indication of the perceived quality of service of VoIP. It is expressed by a number ranging from 1 to 5, 1 corresponding to the lowest quality and 5 to the highest (close humain voice).

MOS Rating	Meaning
5	Excellent
4	Good
3	Fair
2	Poor
1	Bad

Please note that in real network a MOS note of over 4.4 is unachievable. A low MOS will translate into echo and degraded signal. MOS is in principle the result of a series of subjective tests; in the context of network analysis, MOS will be estimated using a formula that integrates 3 factors:

- Network latency (RTT recommended value: <100ms)
- *Jitter* (recommended value: <30ms)
- Packet loss rate (recommended value: <5%)

8.3.4 Prerequisites

To provide MOS values for *VoIP* traffic, it is necessary to capture the three flows: signalization (SIP or MGCP), media (RTP) and control protocol (RTCP). If one of these flows is not present in the traffic capture brought to the listening interface(s), the MOS value will not be calculated. Other quality of service metrics will remain available.

Protocol	Metrics obtained by analysis of the protocol
SIP/MGCP	• Sign. RTT (network latency between each phone – value in & out interval between a request and the first response (definitive or temporary) from the signalization server • Sign. SRT (signalization server response time) • Sign. RD (retransmission delay for the signalization traffic) • Sign. RR (retransmission rate for the signalization traffic) • Code (indicates how the VoIP call ended – e.g. error or not; please note that the code depends on the protocol used)
RTP	• Jitter (standard deviation of latency for the media traffic going from one IP phone to the other) • Packet loss (percentage of packets lost in the conversation at the point of capture of the probe-based on RTP sequence numbers)
RTCP	• RTT (network latency between the two IP phones – based on the timestamps provided by both IP phones)

Note: RTT and MOS values depend to some extent on the quality of the measurement provided by RTCP. Please note that MOS is not very sensitive to "normal" latency values. When referring to voice or media, we refer to the RTP traffic, which may correspond to different things (human voice, prerecorded message, ring back tone, busy line tone, ...) The *VoIP* module discards the *jitter* and packet loss data present in the RTCP flow and replace them with equivalent values computed internally. This is so for several reasons:

- It was observed that many softphones do not place accurate (or even credible) values in these fields,
- RTCP stream is more often missing than present, probably because it is firewalled and of little use to the *VoIP* client software.

For the *VoIP* module to remain passive, there is no other option than compute these values for every RTP stream to generate jitter and packet loss values which will be a good estimate of the real jitter and loss experienced by both users. This is how, even, in the absence of RTCP stream, we can display a jitter and packet loss count (and no RTT, and thus no MOS).

8.3.5 VoIP Views

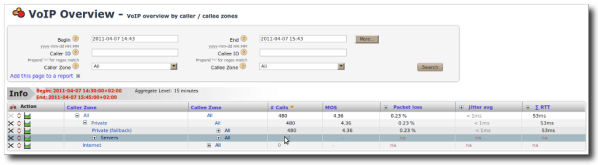
VoIP Overview

VoIP Overview is a view of all VoIP traffic in the network, zone per zone:

- Number of calls
- MOS value
- Packet loss (global or caller / callee)
- Jitter (global or client / server)
- RTT (global or client / server)

Note: The value "caller" / "in" corresponds to the metric for the RTP/RTCP traffic from the caller to the callee and the value "callee" / "out" corresponds to the metric for the RTP/RTCP traffic from the callee to the caller. From each line, you drill down:

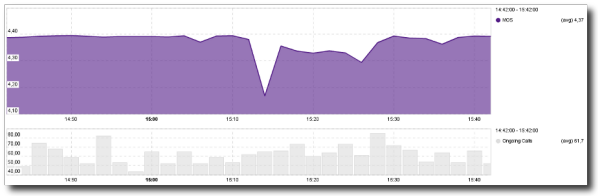
- to the MOS chart,
- to the *VoIP* conversations.



MOS over time

This view shows the evolution of the *Mean Opinion Score* through time. A second graph shows the evolution of the number of calls, to help you evaluate how many were impacted by a MOS degradation.

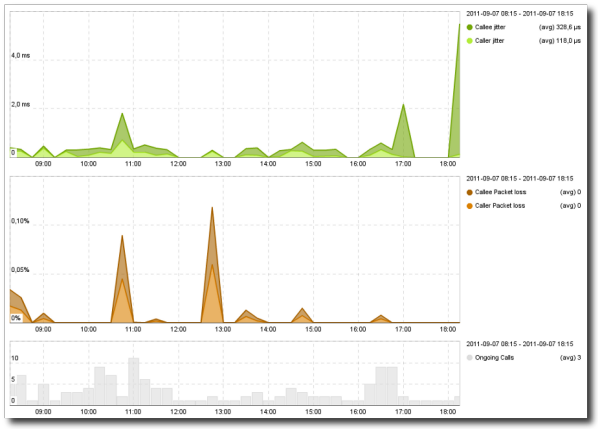
- By pointing a specific point of time on the graph, you can display the exact value for each metric on the right side of the graph.
- By clicking on a specific point of time, you are directly to the *VoIP* conversations for this point of time.



Jitter / Packet Loss

This view shows the evolution through time of the jitter and the packet loss. This view can help you understand MOS variations and see which metric is impacting MOS.

- By pointing a specific point of time on the graph, you can display the exact value for each metric on the right side of the graph.
- By clicking on a specific point of time, you are directly to the *VoIP* conversations for this point of time.



VoIP Bandwidth & Call Volume

This view shows a chart of:

- bandwidth used for voice and signalization for the first one.

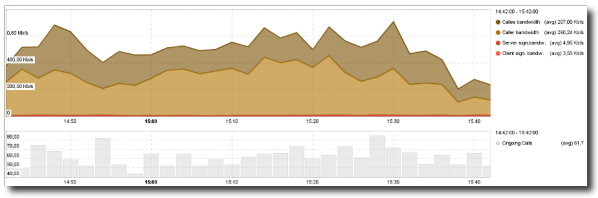


Figure 8.5: VoIP Bandwidth Chart

- the evolution of the volume of calls through time. Calls are distributed between successful and unsuccessful calls. Successful calls are conversations where some voice was exchanged; unsuccessful calls are conversations without any voice exchanged.

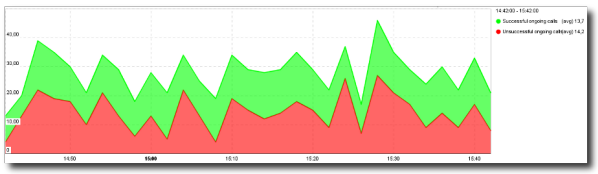


Figure 8.6: VoIP Calls Volume

VoIP Conversations & Details

The two last views show each call individually with some *usage* metrics for *VoIP Conversations*. The *VoIP Details* view is the same table but with *performance* metrics.

8.4 Application dashboards

Dashboard are a report fitting on a single screen that put together all relevant information to understand how the application is doing. They are present in APS from version 1.7.

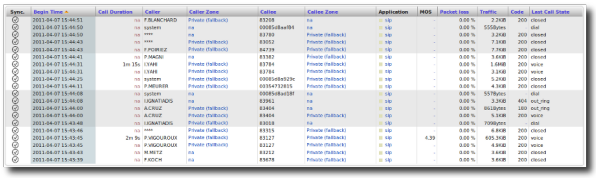


Figure 8.7: VoIP Calls

Note: Those dashboards are not available in *Securactive NPS*.

It is extremely useful:

- as a starting point for troubleshooting,
- as a tool to communicate to management and business users on how the application is actually performing.

It is a set of three elements that display key information on the performance of a business application.

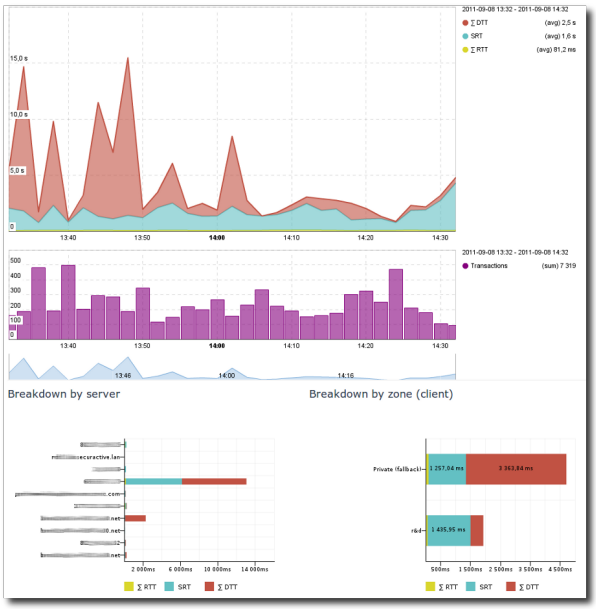


Figure 8.8: Overall view of the application dashboard

8.4.1 How can it help?

For reporting

In a single report you have enough to explain a business user or a manager how the application performance went through time, which servers were doing worse and which zones were impacted. On top of the `EURT`, all this is based on three synthetic metrics that are easy to explain, so that you can address non-technically aware people with an understandable speech about *"what is going on"*:

- `RTT` – network performance
- `SRT` – Server Performance
- `DTT` – Delivery of application response through the network.

For troubleshooting

For network administrators this report brings together all the information about a business application required to:

- validate whether there is a slowdown or not
- identify the origin of a slowdown (network, application, response delivery)
- which users or servers were impacted

In no more than one click, you can conclude on whether there was a slowdown or not, what was the origin of the degradation, which client zones were impacted. With a single additional click (i.e. two clicks in total!), you can view whether all clients in a *zone* were impacted or if the server response time degradation was due to another *application* hosted on the same server machine.

8.4.2 Components

1s element: the evolution of End User Response Time through time

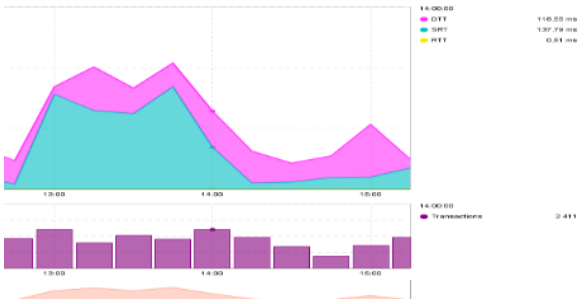


Figure 8.9: End User Response Time (EURT) graph

This `EURT` graph shows:

- the evolution of the quality of experience for users of this application over the period of time,
- the number of transactions help you consider the evolution of EURT with rigor and common sense (you would not consider a degradation of *EU Response Time* for `10` applicative transactions in the same way as for `10 000`).

The breakdown of `EURT` in three intelligible components (`RTT` for network latency, `SRT` for *Server Response Time* and `DTT` for *Data Transfer Time*) let you know at first glance what is the origin of the possible performance degradation. For example in the screenshot here-above, we can observe an increase in the `SRT`; the network and the time required to send the response to the client have not increased. Either the server overall responded slower or some specific queries required a much larger treatment time (you can determine this by drilling down to that specific point of time).

2nd element: EURT by Server

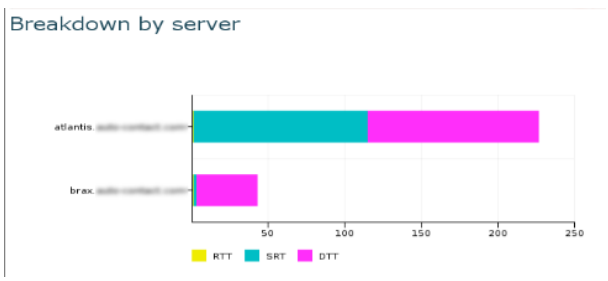


Figure 8.10: EURT by server

What we can see here, is a comparison between the EURT for that application on each server that provides this *application*. In this case, it is obvious that *Atlantis* tend to respond much slower than Brax. By clicking on it having a looking at a second dashboard called *Server/Application Dashboard*, we shall be able to determine if this permanent or punctual and whether this due to the load on this application or on another one hosted on the same server.

3rd element: EURT by Client zone

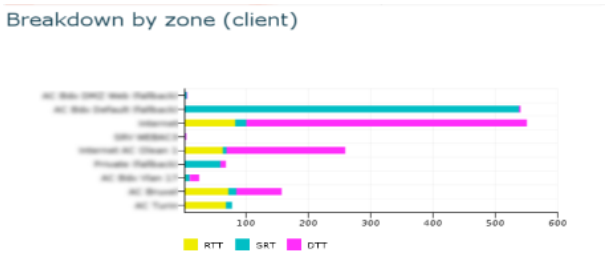


Figure 8.11: EURT by Client zone

What we can see here is a breakdown of the EURT for this *application* between client zones; at one glance, you can determine which *zone* was impacted by the degradation and what are the different level of experienced performance depending on where users are located. For example, from the screenshot here-above, we could certainly think that mainly one zone was impacted by the SRT degradation and also that there are some significant differences in performance between zones due to differences in RTT values (network latency).

8.4.3 Drill down dashboards

SecurActive APS offers two additional dashboards:

- Client zone / application dashboard.
- Server / application dashboard.

Client zone / application dashboard

You can access this dashboard either through the menu or by clicking on a specific client zone in the *Application Dashboard*. This dashboard contains three bits of information:

- EURT graph through time for this client *zone* and this *application*.
- *EURT breakdown by server* (so that you can compare the performance offered by different servers to that client zone).
- *EURT per client* (so that you can identify whether all clients are impacted by a slowdown, or which individual client generates more volume or has worse application performance).

The breakdown by client is interesting to know whether all the zone was impacted or just some individual users and on which component of the EURT (network latency, server response time or data transfer time and for which number of transaction and amount of traffic).

Server / application dashboard

You can access this dashboard either through the menu or by clicking on a specific server in the *Application Dashboard*. This dashboard contains three bits of information:

- EURT graph through time for this server and this application

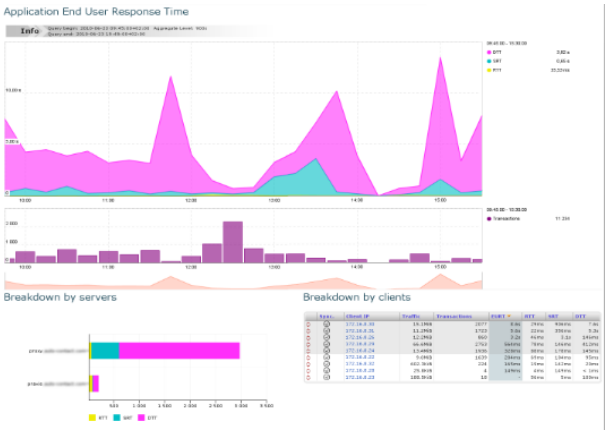


Figure 8.12: Client zone / application dashboard

Figure 8.13: Breakdown by client

- *EURT breakdown by client zone* (so that you can compare the performance offered to different client zone from that server)
- Comparison with other applications provided by that server (so that you can identify whether a peak of transactions on another application is impacting the performance of that application, and see the volume of data, transactions and performance metrics for all applications provided by this server).

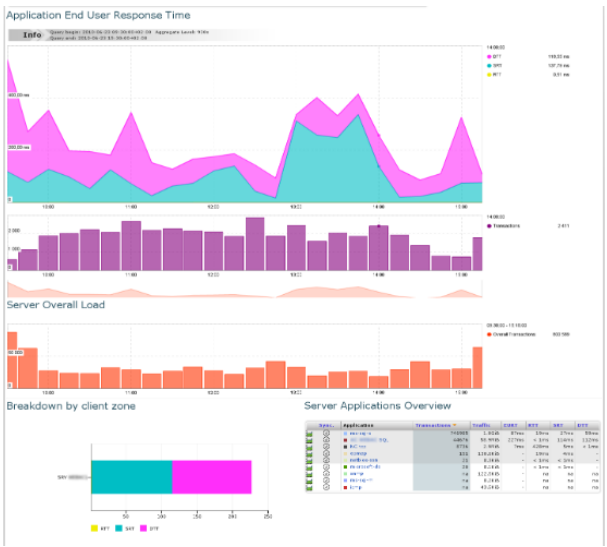


Figure 8.14: Server / Application Dashboard

8.4.4 Interactions

Dashboard have been developed so that a single click drives on more detailed information on the object you are most interested in:

- If you click on the `EURT` graph in any of these three dashboards, you make a focus on a shorter period of

time (for example a SRT peak – depending on the aggregation level you either reach a lower aggregation level for a shorter period or the corresponding performance conversations, see *Data Aggregation*). At the same time you will get the server and zone breakdown for that more specific period of time.

- If you click on a server, you reach the Server / application dashboard.
- If you click on a client zone, you reach the Client zone / application dashboard.

8.5 TCP Errors / Events

8.5.1 Objectives

These two tables expose to the user many TCP statistics in order to reveal dysfunctions or unusual events.

8.5.2 TCP Errors

For each TCP conversation the following fields are displayed:

- RD Server/Client
- Duplicate acks
- number of SYNs
- number of handshakes
- number of session ends
- number of FINs from client
- number of FINs from server
- number of RSTs from client
- number of RSTs from server
- number of timeouts

By sorting on the RD or duplicate ack fields one can quickly check the worst conversations in term of TCP performance. Also, number of reset packets are usually noteworthy. One can then jump to the IP summary page of either the client or the server (depending on who is to blame) to gather further data on this event.

8.5.3 TCP Events

This page does not focus explicitly on TCP errors but aims at giving various overall statistics about each TCP conversation, in order first to give an accurate view of the actual traffic in term of payload and number of connections, and second to notice unexpected patterns.

This page can also serve as a way to find which conversations are important/relevant and thus which zone / application could be split to help distinguish more closely between significant flows.

For each TCP conversation the following fields are displayed :

- payload
- number of packets
- number of handshakes
- number of timeouts
- number of RSTs from client
- number of RSTs from server

- number of FINs from client
- number of FINs from server

8.6 Packet level analysis

8.6.1 Objectives

Once you have identified the origin of an issue, you may want to analyze it further by looking at the packets themselves. You have two ways to realize this:

- *Manual packet capture* through Pulsar's `tcpdump` command
- *Automated Packet Capture ("AutoPCAP")*
- *Triggered Packet Capture* from the data of a result row.

8.6.2 Manual packet capture

By connecting through Pulsar, you can start a manual capture of any traffic viewed on the interface of your device. To do so, you need to go through 3 steps:

1. Connect to Pulsar (see *Pulsar*)
2. Enter the command to launch the trace: for example, `tcpdump_tofile -i <interface> host <host_ip>`.
3. Enter `Control+C` to stop the trace.

Use the `tcpdump` command instead of `tcpdump_tofile` to display the results of real time packet capture.

Note:

- you can access a help by entering `help tcpdump_tofile`.
- you can refer to `tcpdump` command (http://www.tcpdump.org). Please, have a look at the online manual (http://www.tcpdump.org/tcpdump_man.html).
- all parameters are availiable except the `-w`.

Accessing the tracefile

To access the PCAP file generated by the `tcpdump_tofile` command, you should connect to the probe via FTP, using a FTP client and the Pulsar admin user (see *Pulsar*).

8.6.3 Automated Packet Capture ("AutoPCAP")

Principles

Performance Vision can capture packets automatically, in case abnormal values are observed on critical servers. These packets are presented for later analysis as *PCAP* files, which can be downloaded through the web graphical interface at the conversation level.

Applications

These files are presented in the following views:

- Conversations
- DNS messages
- VOIP details

In each of these views, a column at the right end of the table indicates *PCAP*; a small icon indicates that packets have been captured for a given conversation or not. If the *PCAP* file is available, you can download it by clicking on the icon. Once the file has been downloaded, you can view the packets with any protocol decoder (capable of reading *PCAP* files).

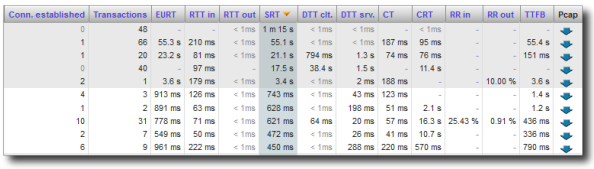


Figure 8.15: PCAP column in Performance conversations

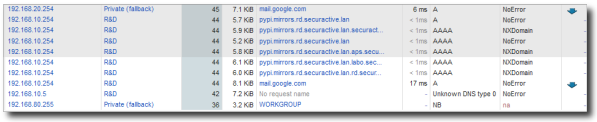


Figure 8.16: PCAP column in DNS messages

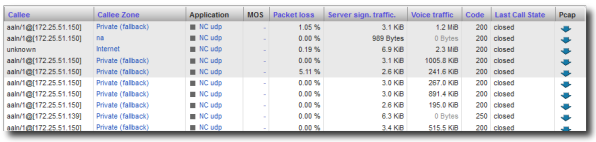


Figure 8.17: *PCAP* in VOIP details

For instance, if you are using *Wireshark* to decrypt the packets, you can directly view the packets.

To view the query and the beginning of the response, you can use the feature *Follow TCP stream* (in the Analysis menu).

Conditions

Packets are saved by Performance Vision, as soon as the conversation they belong to matches a certain number of conditions:

- If *Capture HTTP* if checked in a *Zone* then if an IP address matches the zone subnet (either as client or server).
- If *Capture HTTP* if checked in an *Application* the, if a port or an IP address matches the application (either as client or server).
- And one of the following metrics is considered as out of the norm:
  - Server Response Time `(SRT)` for `TCP` flows
  - Retransmission Rate
  - *DNS* Response Time

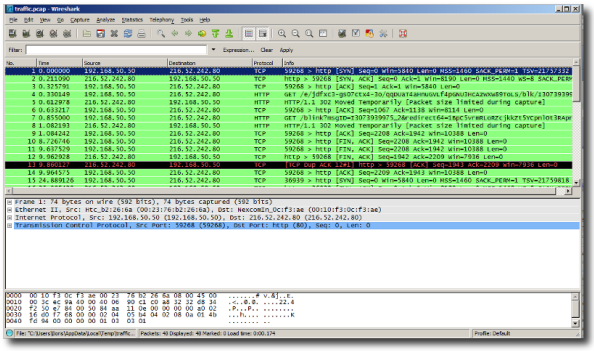


Figure 8.18: Viewing packets in Wireshark

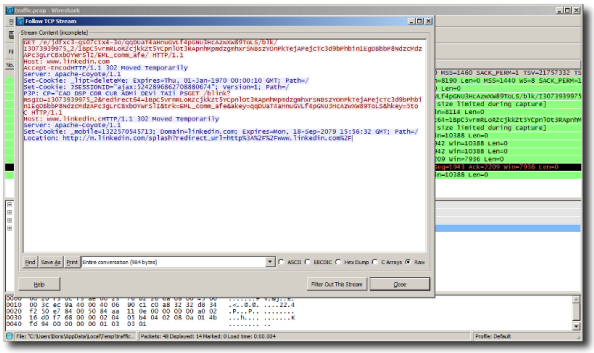


Figure 8.19: Viewing query and response

Note: Why SPV does not use directly a *Zone* or an *Application* to capture PCAP files ?

We want to capture the flow for troubleshooting since the very first packet. But with the information of this only one packet SPV cannot know what is the *Zone* or *Application* of the flow.

Note: *PCAP* files are a **sample** of the conversation. If you request on a one hour interval and get a *PCAP* file, the *PCAP* will not contain one hour of data but only the data which match the above conditions.

Limitations

The Automatic Packet Capture feature works under a certain number of conditions to ensure the proper execution of other services provided by Performance Vision. Among these necessary limitations, you need to observe the following:

- The retention of *PCAP* files is limited by the disk space allocated for captures; in the current version, this space is limited to 10GB by default (for both manual and automatic captures). When all 10GB are used, no new *PCAP* file is saved. You can change this value in “Sniffer Config.” page.
- The maximum retention time for Automatic captures is set to 48 hours; after this delay, Automatic *PCAP* will be deleted. This cannot be modified.
- The sniffer component of Performance Vision is set forge 5 000 *PCAP* files simultaneously; if more than 5 000 conversations are needed, change the parameter in “Sniffer Config.” page, otherwise some conversations will not be recorded at packet level.

Please note that the threshold values and voluntary limitations will be reviewed in newer versions in the light of our experience and the customer feedback we will receive. Please note that if you need an exhaustive trace of a given set of conversations, you can also use the manual capture feature available through Pulsar.

8.6.4 Triggered Packet Capture

Triggered PCAP are generated from the user interface, either by the result rows, or by a configuration page. In both cases, the administrator rights are requested.

The setup is very easy because the capture filters are preset with the wanted flow characteristics, but the main advantage of triggered PCAP is that it is possible to set a date and time to start the capture.

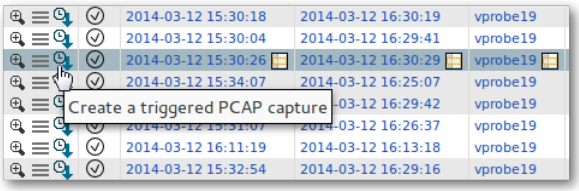


Figure 8.20: Load the form to trigger a new PCAP, the flow data will be used to preset the filters.

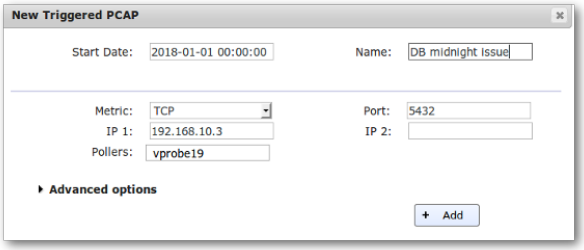


Figure 8.21: Trigger a PCAP for midnight.

By default only the local poller is selected to trigger the capture, but all known pollers are available. If multiple pollers are selected for a capture, then one PCAP will be created for each one.

All added triggered PCAPs are referenced in the dedicated page in the config menu. Is is possible to delete and download them, regardless of the poller where they were captured.

If a capture was done on multiple pollers at a time, then they will have the same name and same filters. They will be grouped together in the management interface.

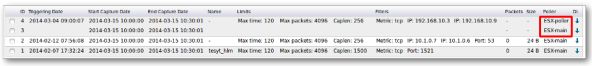


Figure 8.22: The triggered PCAP management page with the first one created on two pollers.

8.7 Interpretation Guidelines

The objective of this section is to help our customers to make the best use of the performance reports provided by their appliance. You will find enclosed a brief overview of how application performance issues can be solved with SPV. This first section focuses on synthetic metrics to produce a measure of the quality of experience of users (QoS - End User Response Time) and give you a simple explanatory framework to understand the cause of application slowdowns (Round Trip Time, Server Response Time and Data Transfer Time).

Note: Some metrics and views described below are only available in *Securactive APS*.

8.7.1 Objectives

Before you start analyzing performance reports, there is a certain number of elements which you must bear in mind: **Performance metrics should not be considered as absolute values, but in comparison with different time intervals, servers and user groups.** Performance metrics represent time interval. Although most of them correspond to the measurement of a concrete phenomenon, it is almost impossible to provide a scale of what is a good or a bad response time, with no experience of the impact it has on users. For example, indicating that the Network Round Trip Time from a `site A` to a `site B` is `200ms` does not mean you have a measure which is acceptable or not. In the same way, a Server Response Time `(SRT)` of an `application A` of `100ms` may be very "bad" when the same value would be excellent for an `application B`. As a consequence, it is important to consider performance metrics as relative values; one of the key to a good interpretation of performance metrics is to compare systematically performance metric value:

- to another time period,
- to another users group.

Mixing up performance metrics for several applications does not make sense. When looking at application performance metrics, you should be very careful of isolating applications for analysis. As a consequence the metrics which very much depend on the application's specific behaviour should not be considered altogether: this is true for metrics such as `EURT` (End User Response Time), `SRT` (Server Response Time) and `DTT` (Data Transfer Time).

RTT measurements can marginally be impacted by the behaviour of the operating system. Network Round Trip Times for `TCP` are based the `TCP` acknowledgment mechanism. This means that, although `RTT` is generally a good measurement of round trip latency, if the operating system of one of the parties is so overloaded that the acknowledgment process becomes slower, `RTT` values will be impacted. RTT Server would be impacted on the server side and `RTT Client` on the client side. `RTT` should then be analyzed in parallel to `CT` (Connection Time - because the treatment of new session by the IP stack has a higher priority).

Some values are averaged measures. For each conversation, two kinds of values are reported:

- counters, for instance packets or byte counters, which are the sum over all connections aggregated for this conversation;
- performance metrics, for instance `RTT`, `SRT`, `DTT` and the likes, which are average values over all samples aggregated for this conversation.

EURT

`EURT` stands for End User Response Time.

This metric is an aggregate of various other measures meant to give an idea of the perceived overall end user experience. It is taken as the sum of `RTT`, `SRT` and `DTT`.

`EURT` has no meaningful physical counterpart. Only its evolution makes sense, and allow the system administrator to check at a glance whether a network *zone* is behaving as usual or not. Notice that expected correct values for both `SRT` and `DTT` depend on the protocol at hand. As a consequence you should not try to compare two `EURT` of different applications.

RTT

`RTT` stands for Round Trip Time.

`RTT` gives an approximation of the time required for a packet to reach its destination, and can be further decomposed into a `RTT Server` (delay between a data packet send by the client and its `ACK` from the server) and a `RTT Client` (in the other way around). As a typical IP implementation will delay acknowledging of incoming data, additional tricks are exploited in order to rule out these software biases :

- make use of `SYN/FIN` acknowledgment and some exceptional conditions such as `TCP` resets, that suffer no such delays, to estimate a realistic upper bound.
- exclude unusually high `RTT` values.

- bound RTT Server/Client by SRT/CRT if RTT sample set looks suspicious.

RTT is meaningful of the bare speed of the physical layer. It is unaffected by packet retransmissions, packet loss or similar occurrences. RTT may be affected by (from most common to the rarest):

- Slow network equipment between client and server (such as a router or a switch);
- Link layer overloaded (ethernet collisions for instance);
- Malfunction of one of the involved network adapter.

These troubles should be further investigated by comparison with other client and/or server zones in order to locate the misbehaving equipment. Notice that a degradation of RTT will almost invariably impact other metrics as well.

SRT

SRT stands for Server Response Time.

SRT gives an estimation of the elapsed time between the last packet of an applicative request and the first packet of the server's response.

SRT represents the processing time of the server, at the application layer, for a given request. SRT may be affected by (from the most common the the rarest):

- Time greedy application request (a complex SQL command can let the server processes during many seconds);
- Application layer overloaded (too many requests, such that the server can't handle all of them in a small period of time);
- Marginally SRT can be affected by the increase of network latency between the point of capture and the server (parallel increase of the RTT Server value);

To pinpoint the root cause of the slowdown, we firstly want to compare the SRT for a given couple server/application to other applications on the very same server. If there is a blatant difference, the application is guilty. Otherwise, we want to compare it to other servers in the same zone, then different zones.

DTT

DTT stands for Data Transfer Time.

DTT server is defined as the time between the first data packet of the response (with ACK flag and a non null payload) from the server and the last packet considered as part of the same response (if the packet has the same acknowledgement number); FIN, RST packets from server or client will also be considered as closing the sequence. A Timeout will cancel a DTT. Note that if the answer is small enough to be contained in only one packet, the DTT will be of '0'.

DTT client is the same metric in the other direction.

DTT (sum of both server and client DTT) is meaningful of the time the user is going to have to wait for the response to circulate on the network from the server to the client. It is not dependent on the Server Response Time (e.g. a DTT might be short for a long SRT:

- the request might require a large calculation, but the result represents a small volume of data; or a DTT might be very large, but SRT very short because the request is easy to handle but the response is very large). DTT depends on (from the largest impact to the smaller):
- the size of the response (the more data is contains the longer it takes to transfer it),
- the level of retransmission (the more packets are retransmitted, the longer it will take to transfer the whole response),
- the network latency (the longer it take to transfer packets through the network, the longer it will be to transfer the response - minor impact),
- the actual throughput which can be reached to transfer the response from the server to the client.

DTT may vary (for most common to a the rarest):

- globally or not on a per transaction basis (if only for some transaction, it may be linked to the size of some specific application response),
- for all client zones or for some only (if for some client zones only, it may be linked to specific network conditions — retransmissions),
- for all servers or for some server (if for a specific server, it may be due to a specific server issue in broadcasting the response).

8.7.2 Scenario guidelines

Slow site connection

Hypothesis:

One or several end users complain about a slow access to all applications (both in and out the LAN).

Diagnosis:

You will find in this section the classical informations to grab in order to diagnose the issue:

- is the application really slower for this site? You can get this information from the Application Performance Dashboard:

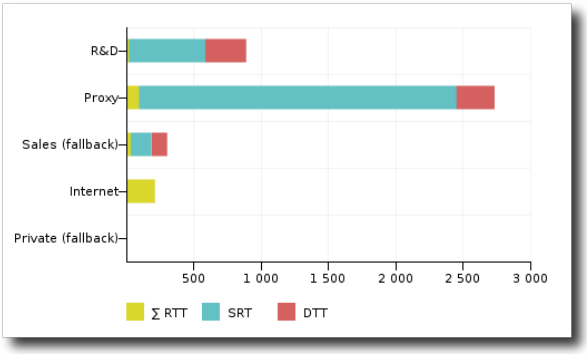


Figure 8.23: Zone comparison in the Application Performance Dashboard

- Does the slowdown occur for a specific application? If so, check *Slow application*.
- Does the slowdown occur for a specific server? If so, check *Slow server*;

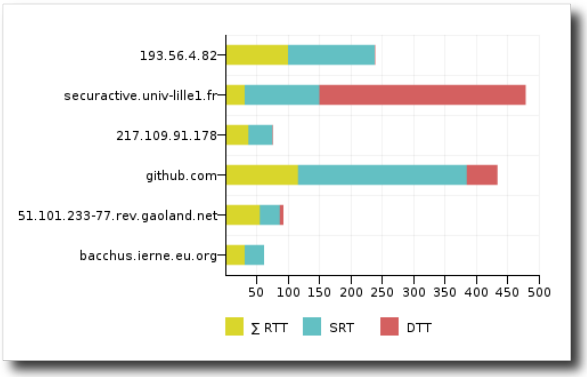


Figure 8.24: EURT comparison between servers in the Application Performance Dashboard

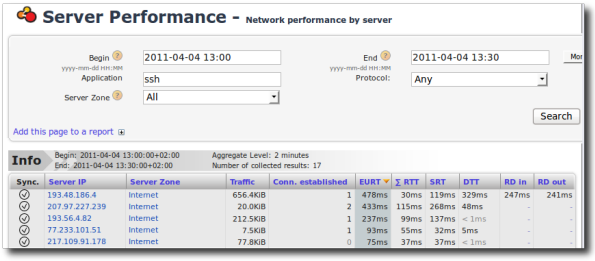


Figure 8.25: Server Response Time comparison through Server Performance.

- Did you upgrade the clients workstations recently? If so, it's a specific systemissue, you may ask the System Administrator for more details;
- Did you upgrade your network equipment? If so, the router/switch configuration is probably involved;
- Now we might inspect deeply in the SPV dashboards. Check the **Monitoring -> Performance Over Time Chart**

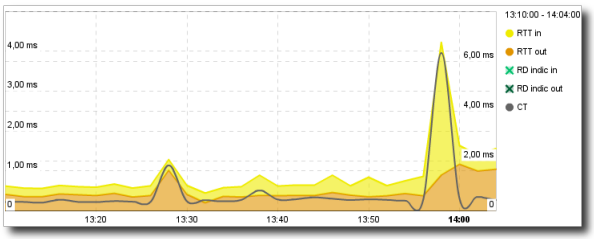


Figure 8.26: Network Round Trip Time analysis

- Do the **Retransmission Rate** and **Retransmission Delay** vary? If so, we might face a congestion issue. Take a look at the router's load, etc;

Figure 8.27: Retransmission analysis

- The general slowdown for a client zone may also be the consequence of a crucial service: the `DNS`. Check out *DNS Response Time*;
- Look at the **Monitoring -> Bandwidth Chart**, to inspect the bandwidth variation, and the number of `TCP/UDP` flows as well.

They might have overcome a QoS threshold, such that all the new application requests are blocked. A hint would be the increasing number of TCP RST packets. To be sure, you may take dive into the **Analysis -> TCP Errors** menu.

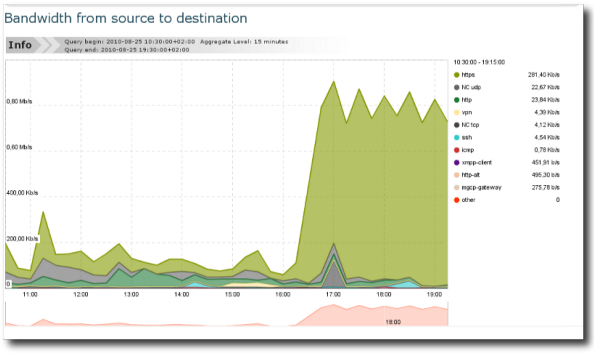


Figure 8.28: Retransmission analysis

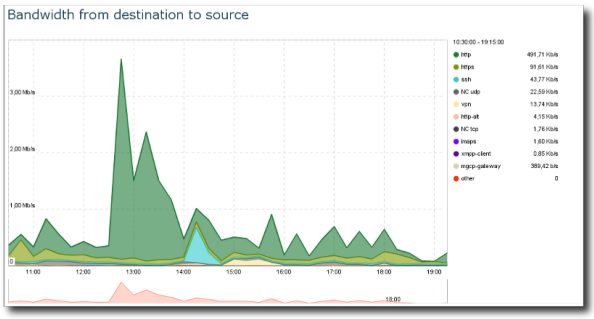


Figure 8.29: Bandwidth charts

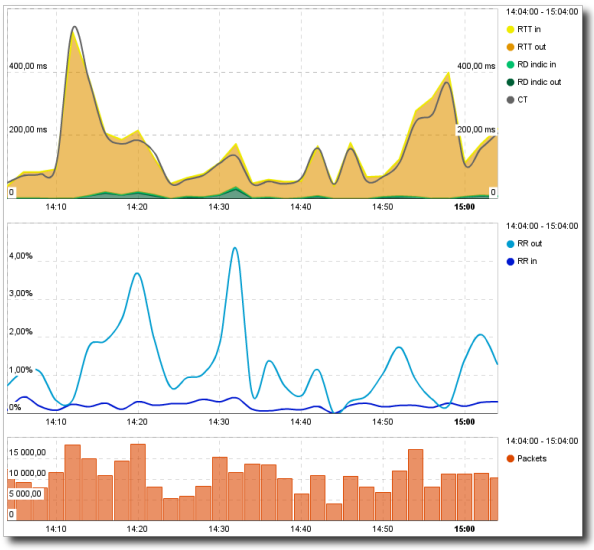


Figure 8.30: Impact of congestion on retransmissions and network latency or connection time

Server IP	Application	Traffic	Packets	Conn. established	Num. timeout	Client RST	Server RST
172.16.1.10	PROXY	6.2MiB	10939	234	47	120	0
172.16.1.12	http-private	1.0MiB	1722	25	9	74	1
proxy.auto-contact.c...	PROXY	19.3MiB	22795	220	32	73	0
proxy.auto-contact.c...	PROXY	9.0MiB	15181	184	44	67	0
172.16.1.10	PROXY	9.3MiB	14615	244	42	61	0
172.16.1.10	PROXY	4.7MiB	8376	208	36	55	0
172.16.1.10	PROXY	11.6MiB	15668	138	26	44	0
172.16.1.10	PROXY	6.6MiB	13134	348	58	43	0
172.16.1.10	PROXY	2.6MiB	4361	99	21	43	0
172.16.1.10	PROXY	3.0MiB	4368	81	16	38	0

Figure 8.31: Number of RST packets sent from the TCP servers

Slow application

Hypothesis

One or several end users complain about a slow access to a specific application : a fileserver.

Prerequesites

Zones have been configured to reflect the customer's network topology. The application `Samba_CIFS` has been identified. The traffic to the fileserver is mirrored to one of the listening interfaces of the probe. Where to start: a global view of the application performance!

1st example

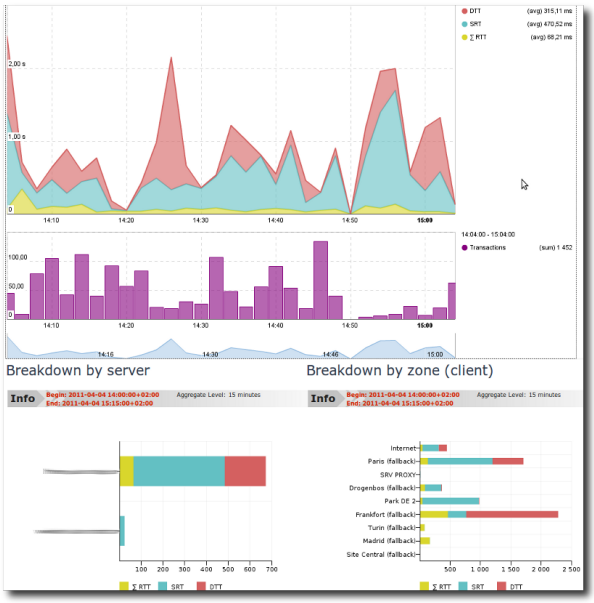


Figure 8.32: Peak in Server Response Time: application performance

Display the Application Dashboard for a relevant period of time. We can easily observe a peak in `SRT` from `6` to `18:15`. From the breakdown by zone, we can easily conclude that only one zone has been impacted.

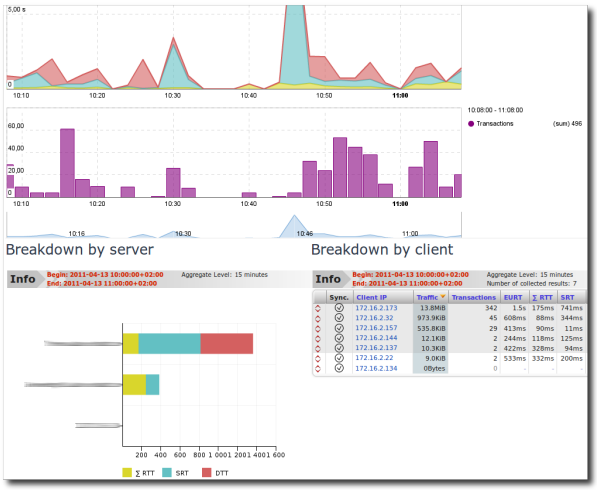


Figure 8.33: Peak in server response time: Application EURT

By clicking on that zone, we can see this client zone application dashboard:

From this, you can conclude that only one client (= user) was impacted. This issue was definitely due to a slow response of the server; it may be due to an application issue or a request which is specifically hard to respond to.

2nd example

Figure 8.34: Peak in server response time: Application dashboard

Application Dashboard for a relevant period in the past (48 hours for example).

This dashboard shows in the upper part the evolution of the End User Response Time `(EURT)` through time for this fileserver.

- We can easily observe that the quality of experience of users accessing to this application got much worse yesterday afternoon.
- We can easily identify that this was due to a degradation of `RTT` (Round Trip Time - indicator of network latency) and not to the Server Response Time `(SRT)` or the Data Transfer Time `(DTT)`.

From this graph, we can conclude that the server and the application are likely not to have any relationship with the slowdown. By looking at the two bar charts which show respectively the breakdown by server and by client zone, we can draw the following conclusions:

- This application is distributed by one server only `(192.168.20.9)`
- The `EURT` vary in large proportion between client zones, mainly because of `RTT`.
- *VLAN_Sales* has a much worse access to the application than *VLAN_R&D*, mainly because of the network latency.

Getting confirmation of our first conclusions. By clicking on the peak of `EURT` in the upper graph, we can narrow our observation period to understand better what happened at that point of time.

This confirms the following conclusions: `RTT` went up for the *VLAN_Sales* (only).

Understanding what is the perimeter of the slowdown

We now know that only *VLAN_Sales* was impacted by this slowdown, due to a longer network `RTT`. We therefore need to understand whether this was general (i.e. impacted all clients in the zone) or isolated to certain clients.

Figure 8.35: Peak of RTT in Application Dashboard

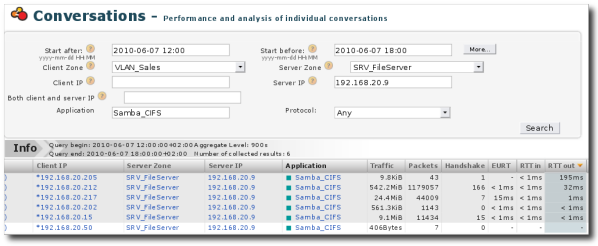


Figure 8.36: Peak in server response time: Conversations

To achieve this, we can simply display the Performance conversations for the application *Samba_CIFS* for the zone *VLAN_Sales*. Here is the result:

From this screen, we can draw the following conclusion:

Only the clients `192.168.20.205` and `192.168.20.212` seem to be impacted. The other clients have very short `RTT` values.

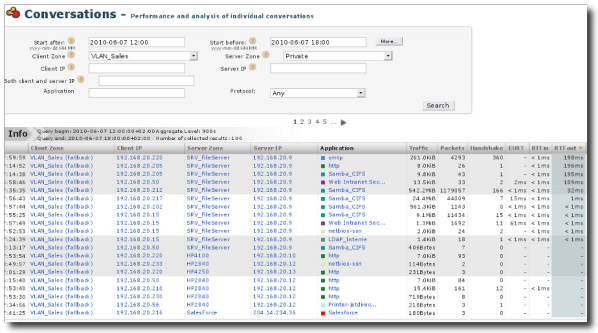


Figure 8.37: Peak in server response time: Conversations

To confirm this, we need to check that these two hosts are the only ones to be impacted and check whether they are impacted only when accessing to the Fileserver. To do that, we have a look at the Performance conversations between the *VLAN_Sales* and the *Private* zone. From this, we can draw the following conclusions:

- Not only `192.168.20.212` and `192.168.20.205`, but also `192.168.20.220` and `192.168.20.50` are impacted.
- The *Samba_CIFS* (access to the fileserver) is not the only application impacted, but `SMTP`, `HTTP` and the *Web Intranet SecurActive*.

Actions to be taken after that analysis

- Check the windowing configuration on the operating system of these hosts (if high value, this is normal).
- Check the level of usage of the host (CPU, RAM usage).

Alternative scenarios:

- If we had seen some retransmission, check whether they are all on the same edge switch and check the interface configuration and media errors.

Slow server

Hypothesis:

Users complain about having to try several times to connect to a web-based application named *"Salesforce"*. The administrator suspects the application server hosting *"Salesforce"* is slow.

How to analyze the problem:

First, check to see if all applications on the application server hosting *"Salesforce"* are slow or if it is just the single web-based application *"Salesforce"* slow. If all applications are slow, then indeed, the application server may in fact be a slow server. If just the one web-based application *"Salesforce"* is slow, while the other applications `(CRM)` are responding quickly, the problem may be the application *"Salesforce"* and not a slow server.

To begin diagnosis, go to "**Monitoring**" -> "**Clt/Srv Table**". Select the application server from the drop-down box labled "**Server Zone**" and click "**Search**".

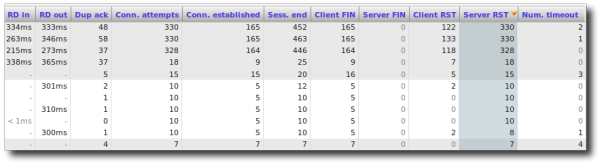
- If we see that all applications on the server are responding slowly i.e. the `SRT` values are high for both *"Salesforce"* and *"CRM"*, the issue related to the server, not to applications.
- Second, check the Connection Time of the application server. If the connection times are high then this may also indicate a slow server.

- Third, check for retransmissions between the clients and the application server. If there are a lot of retransmissions then either the application server or a network device in between are dropping packets. Go to "**Monitoring**" -> "**Performance Over Time chart**". Select the application server "Salesforce" from the drop-down box labled "**Server Zone**" and click "**Search**".

Figure 8.38: Slow server: Performance Over Time chart

Here we see that there is a high Retransmission Rate `(RR Server)` going from the clients to the application server. However, none of the packets from the server to the clients needed to be retransmitted (`RR Client` is around `0`). This indicates that the application server is in fact dropping the packets and is therefore a slow server (Assuming that the route taken form the client to the server is the same route taken from the server to the client as is industry standard practice).

Lastly, check the *TCP errors* of the clients and the *Application server*. If the server reset count or number of timeouted sessions is high, this is a further indication of a slow server. Go to **Analysis -> TCP errors**. Select the application server *"Salesforce"* from the drop-down box labeled "**Server Zone**" and click **Search**.



RD in	RD out	Dup ack	Conn. attempts	Conn. established	Sess. end	Client FIN	Server FIN	Client RST	Server RST	Num. timeout
334ms	333ms	48	330	165	452	165	0	122	330	2
263ms	346ms	58	330	165	463	165	0	133	330	1
215ms	273ms	37	328	164	446	164	0	118	328	0
338ms	365ms	37	18	9	25	9	0	7	18	0
-	-	5	15	15	20	16	0	5	15	3
-	301ms	2	10	5	12	5	0	2	10	0
-	-	1	10	5	10	5	0	0	10	0
-	310ms	1	10	5	10	5	0	0	10	0
< 1ms	-	0	10	5	10	5	0	0	10	0
-	300ms	1	10	5	10	5	0	2	8	1
-	-	4	7	7	7	7	0	0	7	4

Figure 8.39: Slow server: TCP Errors

Here we see that there are a lot of server resets and timeouts. Given all the above information, we can conclude that the application server is operating slowly. At this point, the server administrator should perform direct diagnosis on the application server to verify CPU, RAM and HD usage.

N-tier application performance issue

Hypothesis:

Users are complaining about slow response time from an in-house web application. This application being an N-tier architecture, its performance as seen by a client is tied to several parameters:

- DNS latency to resolve web server name from the client host (see **DNS Response Time**)
- Connection time to server
- Data Transfer Time between these hosts
- DNS latency to resolve other server names accessed from the web server (database servers for instance, cf. **DNS Response Time**)

- Connection and data transfer times between these hosts
- Server response time of these servers

Identification of the culprit:

First we need to find out if the experienced slowdown is due to the web front end itself. To this end, check every component of the EURT:

- If SRT is fast but RTT and/or DTT (see also Connection Time) then we are facing a network slowdown. Refer to previous sections of this guide to further track down the problem.
- If SRT is preponderant compared to DTT and RTT then the application itself is to blame. Proceed to find out what is affecting performance.
- Then check EURT between web server and each other involved servers (databases...)

If some of these EURT appear to be degraded then check recursively these other hosts. If not then check the web server load average.

8.7.3 Additional metrics

TCP anomalies

RST packets

A TCP connection is reset by a RST packet. There is no need to acknowledge such packet, the closure is immediate. A RST packet may have many meanings:

- If a TCP client tries to reach a server on a closed port, the server sends a RST packet. The connection attempt could be a malicious one (port scanning – nmap, etc), or the consequence of an unexpectedly down server, client/server misconfiguration, server restart, etc;
- A router might send a RST packet if the incoming TCP packet does not fit with the security policy (source range IP address is banned, the number of connection attempts is too high in a small period of time, etc);
- A QoS (Quality of Service) equipment limits the bandwitdh (or the number of connections) by sending a RST packet to any new connection attempt;
- If a Intrusion Detection System (e.g. Snort) detects a malicious connection, he can send a RST packet to roughly close it;
- If a host between Client and Server wants to do a Denial of Service, it can reset the connection by sending RST to both peers. Basically it's the same mechanism than the previous one, but the motivation is quite different.

Retransmissions

One of the TCP metrics which is interesting to analyze is the retransmission. A TCP Retransmission is when a TCP packet is resent after having been either lost or damaged. Such retransmitted packet is identified thanks to its sequence number. In *SecurActive SPV* we do not consider packets with no payload, since duplicate ACKs are much more frequent, and not really characteristic of a network anomaly. There are several common sources of TCP retransmission:

- A network congestion. If a router can't cope with the whole traffic, its queue will grow bigger until it gets full and then start dropping the incoming packets. If you reach a predefined QoS limit, the exceeding packets will be dropped as well. Such drop will result in TCP retransmission. A common way to identify this kind of problem is by taking a glance to the traffic statistics. If you see a flat line at the max traffic allowed, then you get the root cause of retransmission. If the traffic graph looks OK, you can check over

the load of the routers/switches you own (e.g. with the SNMP data). If the load is too high, you found the culprit.

- An overloaded server. Check the **Section Slow Server**.
- A hardware failure. Maybe a network equipment is simply down. It will obviously result in TCP retransmission until a new route is computed, or the issue fixed. This type of retransmission should occur with very short time effects and give some quite big peaks of retransmission, on very broad types of traffic on a specific subnet. If this happens often, it becomes important to find the faulty hardwares by tracking down which subnets are concerned.
- A packet header corruption. Network equipments are used to rewrite portions of packets (Ethernet source/destination, IP Checksum, maybe TOS field). A buggy firmware can result in corruption while rewriting protocol headers. In this case, the packet will probably be dropped within the network route. Even if it reaches the destination, the TCP/IP stack won't consider it as a valid packet for the current TCP sessions, and the stack will wait the correct packet. It will end in a TCP retransmission, anyway. This problem will likely occur on the same type of traffic and continuously.

ICMP

What is ICMP?

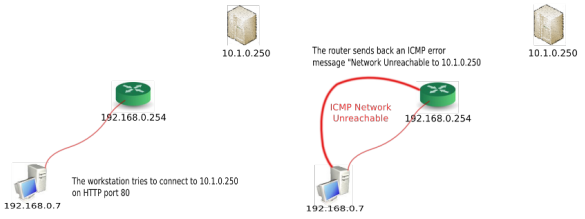
ICMP stands for *Internet Control Message Protocol* and is also a common IP transport protocol. It seems pretty explicit, although most people reduce ICMP to ping reply commands, a good way to test whether a host can be reached through a network and how much it takes for a packet to make a round trip through the network... Obviously ping and trace-route-like tools are very useful for network administrators... but there is much more to say about ICMP and the help it can provide for network administration & diagnosis. In total, ICMP can be used to send more than twenty types of control messages. Some are just messages, some others are a way for IP devices or routers to indicate the occurrence of an error.

Error messages

Let's describe the most typical ICMP error messages you can find on networks.

ICMP Network Unreachable

Let's take the simplest example: one machine sitting on a LAN (192.168.0.7), has one default gateway (192.168.0.254), which is the router. It is trying to reach a server, which does not sit on the LAN (10.1.0.250) and which cannot be reached, because 192.168.0.254 does not know how to route this traffic.

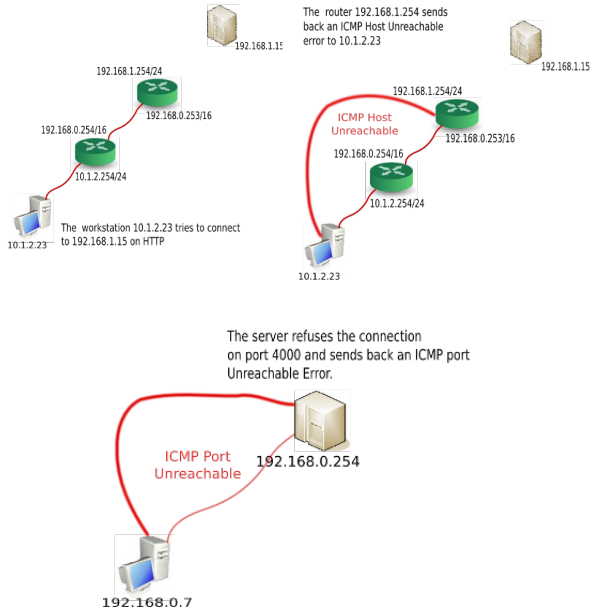


ICMP Host Unreachable

Let's take the simplest example: one machine sitting on a LAN (10.1.2.23), has one default gateway (10.1.2.254/24), which is the router. It is trying to reach a server, which does not sit on the LAN (192.168.1.15). The traffic flows and reaches the last router before the server (192.168.1.254/24); this router cannot reach 192.168.1.15 (because it is unplugged, down or it does not exist).

ICMP Port Unreachable

Let's take a second example: one machine sitting on a LAN (192.168.0.7). It is trying to reach a server 192.168.0.254, which sits on the LAN on port UDP 4000, on which the server does not respond.


The router 192.168.1.254 sends back an ICMP Host Unreachable error to 10.1.2.23
192.168.1.15
192.168.1.254/24
192.168.0.253/16
192.168.0.254/16
10.1.2.254/24
The workstation 10.1.2.23 tries to connect to 192.168.1.15 on HTTP
10.1.2.23
ICMP Host Unreachable
The server refuses the connection on port 4000 and sends back an ICMP port Unreachable Error.
ICMP Port Unreachable
192.168.0.254
192.168.0.7

Where is the challenge with ICMP?

You may be tempted to say: if it is that simple, why do we need SecurActive SPV on top of any sniffer? All the information sits in the payload. But in every network, you will find some `ICMP` errors... they may be due to a user trying to connect to a bad destination, or trying to reach a server on the wrong port. The key is in having a global view of how many errors you have normally and currently and from where to where. The key to leveraging `ICMP` information is in having a relevant view of it and understanding what it means.

How can ICMP help on network diagnostic and security monitoring?

From the explanation here above, we can keep in mind that by analysing `ICMP` errors we can identify machines that try to connect networks or machines, that are routable from the LAN's machine or ones that try to connect on actual servers but for services which ports are not open. Here are some examples of phenomena that can be identified that way:

Misconfigured workstation

A workstation repeats a large volume of missed attempts to connect to a limited number of servers: it may be that this machine does not belong to the company's workstations (external consultant on the network, whose laptop is trying to reach common resources on his home network -DNS, printers,...), or it may be the machine of someone coming from a remote site with its own configuration or a machine that has been simply wrongly configured.

How would we see it?

A large number of *ICMP Host Unreachable* errors coming from one or several routers to this machine or this group of machines. The `ICMP` information contained in the payload of each of these errors would probably show they are trying to reach a certain number of hosts for some services or applications.

Migration legacy

A certain number of machines keep requesting DNS resolution to a `DNS` server which has been migrated (this could be true for any application available on the network). Their users certainly feel worse performance when trying to use these services.

How would we see it?

A large number of *ICMP Host Unreachable* errors coming from one or several routers to a group of machines. The ICMP information contained in the payload of each of these errors would probably show they are all trying to reach the previous IP address of a given server.

Network device misconfiguration

A router does not have a route configured; some machines are trying to reach some resources, unsuccessfully.

How would we see it?

A large number of ICMP Network Unreachable errors coming from one router to many machines. The ICMP information contained in the payload of each of these errors would probably show they are all trying to reach the same network through the same router.

Port scanning

A machine is trying to complete a network discovery. It is trying to connect to all servers around to see on which ports they are open.

How would we see it?

A large number of *ICMP Port Unreachable* errors coming from one or several routers corresponding to a single machine (the one which is scanning).

Spyware / Worms

An infected machine is trying to propagate its spyware, virus or worm throughout the network; obviously it has no previous knowledge of the network architecture.

How would we see it?

A large number of *ICMP Host Unreachable* errors coming from one or several routers corresponding to a limited number of hosts, trying to reach a large volume of non existing machines on a limited set of ports.

Server disconnected/reboot

A service on UDP (DNS, Radius...) is interrupted because the server program is temporarily stopped or the host machine is temporarily shutdown. Many requests are then discarded.

How would we see it?

Many *ICMP Port Unreachable* messages (preceeded by some unreachable host if the host itself was shut down) are emmited during a short period of time for this service host/port.

DNS Response Time

Background:

The DNS (Domain Name System), which has been defined in detail in the **RFC 1034** (http://tools.ietf.org/html/rfc1034.html) and **RFC 1035** (http://tools.ietf.org/html/rfc1035.html), is key to the good performance of TCP/IP networks. It works in a hierarchical way; This means that if one of the DNS servers is misconfigured or compromised, all the network, which relies on it, is also impacted. Although the

DNS protocol is quite simple, it generates a significant number of issues: configuration issues, which affect the performance of the network as well as security issues, which jeopardize the network integrity. The purpose of this section is to cover the main configuration issues you may encounter with DNS when it comes to network performance.

Hypothesis:

You noticed a general slowdown for a specific host, zone, or the entire LAN. You didn't find out the issue with the previous methods. Maybe this problem has nothing to do with the business applications or you network equipment.

Diagnosis:

The DNS server(s) need to have a very high availability to resolve all the names into IP addresses that are necessary to good function of applications on the network. An overloaded DNS server will take some time to respond to a name request and will slow down all applications, that have no DNS data in their cache. An analysis of the DNS flows on the network will reveal some malfunctions like:

Latency issues

If we can observe that the mean time between the client request is significantly higher than the average (on a LAN it should remain close to 1 ms), we may face three kinds of issue:

- the client is not requesting the correct DNS server (DHCP misconfiguration, for example). You can check this out in the interface by looking at the **Server IP** fields;
- it means that the DNS server has an issue with regards to the caching of DNS names. The cache system makes it possible to resolve a name without requesting the DNS server, which has authority for the DNS zone, the IP address corresponding to the name. Hence, if the response time is high, first the application will be slow from the user's point of view, and secondly it will incude an unnecessary consumption of bandwidth. This bandwidth will be wasted both on the LAN and on the Internet link (if we make the hypothesis that the authority server sits on the Internet). If we consider the case of a fairly large organization, the bandwidth used by the DNS traffic will not be negligeable and will represent an additional charge;
- the DNS server may have system issues. If the server is overloaded, it cannot hold all the requests, and delay (or drop) some, which leads to a general slowdown of the network perfomances.

You can easily cast a glance at these issues: go in the **Analysis -> DNS Messages** menu, and fill the form with appropriate values (especially the **Requester Zone**), to verify if the requests are correctly answered, and in an acceptable timing.

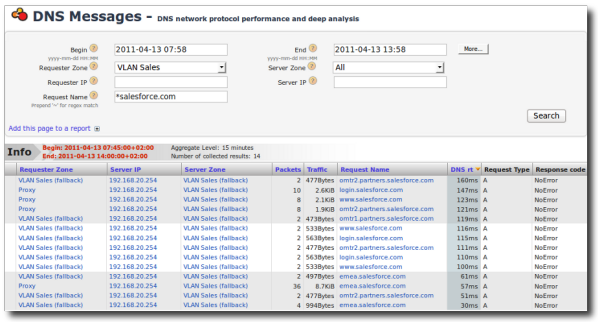


Figure 8.40: DNS Response Time for a specific requester zone (here, VLAN_Sales)

Traffic issue

If we establish the top hosts making DNS requests, it will be possible to pinpoint misconfigured clients, not keeping in a local cache the DNS server responses; this approach makes it possible to distinguish between an issue coming from the user's workstation and one coming from the general function of the network. Please note that hosts making a very high volume of DNS requests may correspond to a malicious behaviour; for example, some malwares try to establish connections to Internet by resolving domain names and sometimes the DNS protocol is used in cover channels to escape information.

DNS errors issue

We can also ask for the top hosts receiving most DNS error messages (non existing hosts, etc.). This will also put the light on misconfigured stations, generating an unnecessary traffic and lowering the overall network performance.

DNS Internal misconfiguration

To do this, we need to identify the AXFR and IXFR transactions towards its autorithy server. If these updates occur too often (and therefore generate an unnecessary traffic), we can conclude that there is an issue. If the bandwidth used is too large, it means that our DNS server requests a full zone transfer (AXFR) when an iterative transfer (IXFR) would have been more adequate. If this is the case, then the network administrator can take some easy steps to improve his network's performance.

CHAPTER

NINE

LICENSING AND UPGRADES

9.1 Performance Vision

Covers all Network, Application and VoIP performance related aspects, and allows to:

- Provide clear information on the mapping of traffic.
- Continuously analyze network and application usage.
- Improve configuration and optimization of IT infrastructure.
- Proactively manage network capacity to avoid congestions.
- Identify opportunities to make infrastructure savings.
- Measure the Quality of Experience of end users vs. SLA.
- Diagnose performance degradations and accelerate resolutions.
- Identify slowdowns, their origin (network, server, application...) and their impact.
- Manage performance of complex application chains.
- Analyze the impact of application deployments on network resources and end user performance.
- Perform deep transaction analysis at application level for major name services, web, database and file sharing protocols.
- Get a full view of performances in both hardware and virtual based environments.

9.2 Deployment Mode

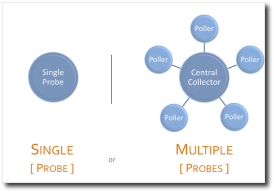


Figure 9.1: Deployment Mode

9.2.1 Single Probe: Stand-alone Appliance

Performance Vision in a stand-alone mode is composed of a single unit which analyzes the traffic, stores the statistics and presents the data through an interface.

9.2.2 Multiple Probes: Distributed Architecture

Performance Vision in a distributed mode will capture and analyze traffic in several physical locations through distinct appliances called “Pollers”. They can be either physical or virtual appliances, which send their statistics to a central Performance Vision unit, called “Collector”. All the data is aggregated into a single database and accessible through a single User Interface.

9.2.3 Platforms

Performance Vision probes are available for two different platforms:

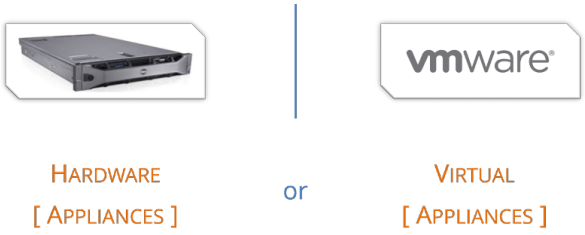


Figure 9.2: Performance Vision Platforms

Hardware Appliances

This product line is based on hardware appliances, specifically tuned for high performance rates. The range of appliances available makes it possible to provide an adequate solution for all size of situations.

Virtual Appliances for VMWare systems

This product line is designed for an implementation in virtualization servers in VMWare systems. The range of appliances available makes it possible to provide an adequate solution for all size of situations depending on the resources (CPU, memory and disks) allocated to the virtual instances.

9.3 Product Range Summary

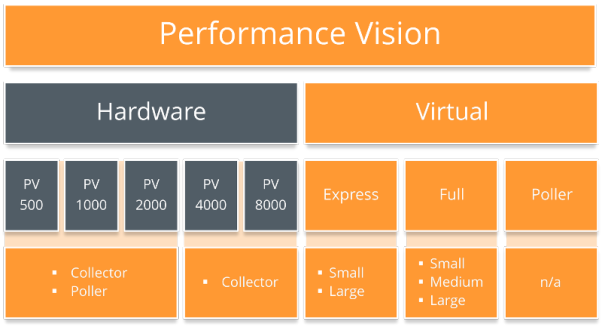


Figure 9.3: Product Range Summary

Our product range is summarized in the following way:

1. Hardware Appliances (1U & 2U Servers)
   - Performance Vision: all features
   - Poller: remote probe for distributed environments

2. Virtual Appliances (for VMWare environments)

- Full (Small Medium or Large): all features
- Evaluation: for testing
- Audit: for auditing services
- Express (Small or Large): entry level
- Poller: remote probe for distributed environments

9.4 Hardware Versions

The hardware appliances comes only in Full or Poller versions:

Full enables administrators to monitor all network, application and VoIP aspects of large datacenters, including redundant ones and dealing with multiple locations through a distributed architecture. Depending on the amount of traffic to be analyzed, here are the flow analyses recommendations for the different versions of the hardware appliances :

- PV-500 < **40,000** flow analysis.
- PV-1000 < **80,000** flow analysis,
- PV-2000 < **200,000** flow analysis,
- PV-4000 < **300,000** flow analysis,
- PV-8000 < **400,000** flow analysis.

Please, see Licensing Model chapter for more details.

Poller is a specific version for distributed environments that acts as a remote analysis point. One or several pollers can be installed on different locations. It works in conjunction with a central collector which must be a Performance Vision Full as only this version supports distributed environments.

9.5 VMWare Versions

The Performance Vision virtual appliances comes in several versions in order to fit different customer's needs.

- The Free version allows you to have some basic overview of the network level aspects on your traffic. Obviously, advanced features, application level, reporting, import/export features are not available, but... it's free for one year!
- Evaluation is a free evaluation version that allows you to test all features for 15 days! After that period of time, it will automatically switch to the Free version.
- The Audit version is dedicated to our partners. With this version, valid during 15 or 30 days, they can provide value added services to customers like performance assessments or in-depth usage audits.
- Express is an entry-level version designed for small networks management and for small network assessment or audits. For a very affordable price, it enables administrators to control network, application and VoIP usage & performance, even if some of the advanced features are not available.
- Full version enables administrators to monitor all network, application and VoIP aspects of large datacenters, including redundant ones and dealing with multiple locations through a distributed architecture. Depending on the amount of traffic to be analyzed, the Full comes in three different versions:
  - Small: < **20,000** flow analysis,
  - Medium: < **100,000** flow analysis,
  - Large: < **500,000** flow analysis.

9.6 Performance Vision Versions

Please find below the detailed specifications of the different Performance Vision versions.

	Free	Evaluation	Audit	Express	Full	Full	Poller
	Virtual	Virtual				Hardware	Both
Data History	15 days	n/a	n/a	15 Days	360 days	360 days	n/a
Expiration	1 year	15 days then Free Version	15 / 30 days	n/a	n/a	n/a	n/a
Max. Flow Analyses	500K	500K	500K	20K, 500K	20K, 100K, 500K	n/a	n/a
Distributed	-	-	✓	-	✓	✓	n/a
Reporting	-	✓ watermark	✓	-	✓	✓	n/a
PDF Export	-	✓ watermark	✓	✓	✓	✓	n/a
CSV Export			✓		✓	✓	n/a
SNMP	-	✓	✓	-	✓	✓	✓
Config. Import/Export	-	-	✓	✓	✓	✓	n/a
Access to Support	-	-	-	Option	✓	✓	✓

Figure 9.4: Performance Vision Versions

9.6.1 Licensing Model

The licensing model is based on the capacity on the central database. Whatever the number of pollers: only one local poller, or several remote ones, what is taken into account for the sizing, is the amount of data processed by the central collector.

There is only one criteria: the central database capacity in terms of numbers of flow analysis. Four versions exist in the Performance Vision product range, they offer the following capacity levels:

Hardware

- PV-500: < **40,000** flow analysis (recommendation),
- PV-1000: < **80,000** flow analysis (recommendation),
- PV-2000: < **200,000** flow analysis (recommendation),
- PV-4000: < **300,000** flow analysis (recommendation),
- PV-8000: < **400,000** flow analysis (recommendation).

Virtual Express

- Small: < **20,000** flow analysis,
- Large: < **500,000** flow analysis,

Virtual Full

- Small: **< 20,000** flow analysis,
- Medium: **< 100,000** flow analysis,
- Large: **< 500,000** flow analysis.

9.7 How can I determine the model that is right for me?

The simplest way is to deploy the *Evaluation* version. You will then find a dedicated screen called *Database Workload* located in the *Configuration* area. It displays the number of different flow analyses integrated in the database over the time. So it is easy to determine, at a glance, what version fits best your needs depending on your specific traffic.

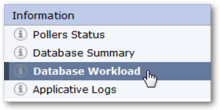


Figure 9.5: Database Workload in the Configuration area

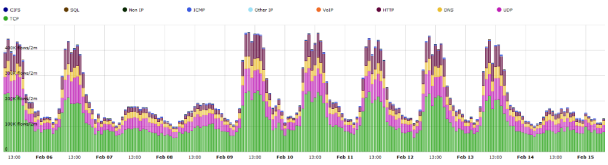


Figure 9.6: Chart of the number of flows analyzed

9.8 Limits

Each Performance Vision version supports a given number of flow analyses. One flow analysis is either a set of exchanges between one client and one server for one application or a layer 7 transaction.

If the data processed by the central collector reaches the license limit, the system will still continue to work smoothly. An email alert will be sent to the administrator to make him aware that the limit has been reached. All data above the limit will be ignored by sampling and will not be processed. In such case, the result is that only a part of the total traffic will be analyzed.

Whatever the Performance Vision version (2015 licenses) there is a protection at **500,000** flows analyses to maintain the stability and efficiency of the system. All flow analyses made over this limit will be ignored and will not be processed. In such case, the result is that only a part of the total traffic will be analyzed. An alert is sent to the Administrator.

9.9 License and Upgrade Installation

Apart from the trial version, all virtual and physical appliances are provided with no license key. You have to get the license key, which will be provided by email by SecurActive.

All SPV entities: virtual, poller and collector (see *Distributed Architecture*) **needs** a specific license. The licenses are specific to a given hardware serial number (the *device id*), so that each device must be sent its own license package.

Note: With this system you can transform an `APS` into an `APP` or an `APS Free` into an `APS express` for example. There is a special case, when you transform any `APP` probe into a `non-APP` you must do a **format_data_disk** command after installation to be able to save captured data. (See *Pulsar*)

The same procedure must be performed for all the entities either for **license** or **upgrades**, please follow the steps below:

1. Connect to the *FTP* server of the probe

   ```
   user: ftp
   password: S3c7r!
   ```

2. Upload (`put`) your license or upgrade file.

Wait a few minutes and it's done! The installation is complete when the license key is not available anymore by refreshing the destination folder lists.

Check your license or new version with the `status` or `poller` commands. For upgrades, please redo the same procedure on all the entities.

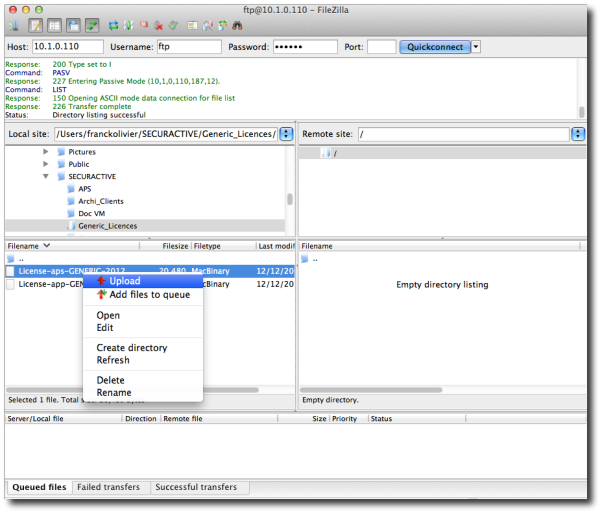


Figure 9.7: Filezilla uploading a license file.

Warning: It is STRONGLY recommended to reboot all the probes after upgrading (use the `reboot` command in *Pulsar*).

Note: Security

The FTP access is writable only (no read). It allows only to put a Securactive signed and encrypted file. This file will be automatically moved, checked and executed by an internal process.

ServicePack

In rare cases, it's needed to upgrade some third-party internal softwares. The information is available in the release note of the new version. These packages are called Service Packs. To apply them, put the file (`SPV-ServicePackX-rY.bin`) using the same method.

9.9.1 Check the license or upgrade

The status of the license can be validated in *Pulsar* with the command "poller".

```
Usage:
  poller
  poller add IP
  poller modify IP
  poller delete IP
APS# poller

Name:            APS
Address:         localhost
Created the:     2012-02-10 17:23
Device ID:       564D9B2D-C67F-A562-730B-6848A685068Z
Device md5:      89fa72d1020bafe6fcc00da13b364d2a
Time:            2012-02-15 17:24
SPV Version:     2.5.13-r2
Sniffer status:  ok, pid 2760
Sniffer version: 2.5.13
License:         invalid
Expiration:      no limit
```

Figure 9.8: Pulsar: command 'poller'

It can also be done through the web interface, in the page "Poller status" in the *Configuration* section. The page displays `SPV version` and `License Status` and then see if your upgrade or license is correctly installed.

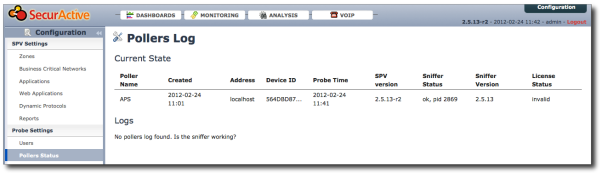


Figure 9.9: Poller Status page: invalid license

CHAPTER

TEN

FREQUENTLY ASKED QUESTIONS

10.1 Firefox freezes randomly on some pages

This seams to be caused by the java plugin, and deactivating this plugin fixes the issue. This has no effect on SPV since it does not use java. To disable the Java plugin, enter the *Tools* → *Add-ons*. This will open a new window with a button bar on top, with a *Plugins* icon. Select it, and it will open the list of all currently installed plugins.

Locate your java plugin, that is the one that handles java applets (on the following screenshot it's titled *IcedTea NPR Web Browser Plugin*, but it may also appear under the name *OpenSDK*, or merely Java). Once located, select it and click on the *Disable* button. You should then restart firefox.

Note: This should not appear anymore since release 2.10, Flash plugin is not required anymore.

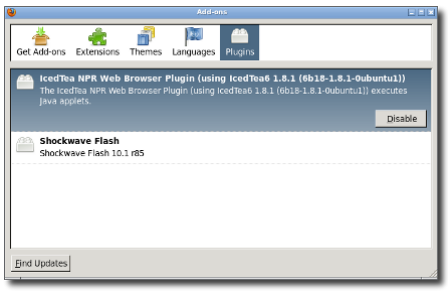


Figure 10.1: The Add-ons pop-up window of Firefox

10.2 Aggregate level changes when browsing from tables to charts

The aggregate level for tables is chosen to display a synthetic view on data, while the charts choose the aggregate level in order to have enough points to plot. So, this is not an error if the aggregate level changes from one page to another.

10.3 How can SRT be greater than DTT ?

Every `DTT` is preceded by a `SRT` but both are not computed simultaneously:

- `DTTs` are not stored until the data transfer is complete;
- `SRTs` are stored as soon as the first packet of the response is seen.

Thus it is frequent to have more `SRTs` than `DTTs` when browsing recent data.

10.4 How can we have 0 packets and no traffic at all on a conversation?

This is a common case when the observation period encompass the end of a timeouted conversation. No packets have been sent during the observation period and the elapsed time since last packet have reached the timeout limit.

10.5 What is this timeout column (in Analysis/TCP Error)?

As there are no timeout in standard protocol (as `TCP`, `UDP` ...) this is an application level notion that the packet sniffer must guess. We consider the conversation as timeouted after `2` minutes without packets exchanged.

10.6 Why are some DNS request names missing?

Although DNS protocol states that the question section must be present in the requests, not all DNS messages are name resolution requests. Some DNS server may use message types unknown of the traffic analyzer that do not embed anything meaningful in the question section of the message. For instance, the NBNS server statistic report is such a message that makes no use of the question section.

Note that you can search for empty DNS names using the regular expression `~^$` in the name search box.

10.7 Some TCP conversations are reported twice, what's wrong?

First make sure that the deduplication process is not configured too tightly. If the faulty TCP conversations keep being reported twice then maybe the duplicated packets are altered in some way that makes them too different from the originals. For instance, some firewall randomize the ISN (Initial Sequence Number) of TCP connections (for security reason). So if you mirror some traffic before and after passing though such a firewall this traffic will be reported twice since their sequence number will be different.

10.8 Pcap files generated by tcpdump are (mostly) empty

By far the most probable reason for this is that you are trying to use a filter on VLAN tagged packets. This won't work since *Tcpdump* filters look for fixed locations in the packet and the VLAN tag offsets the actual bytes that are being matched. Fortunately there is a workaround: by adding the filter `vlan` all following filters will be offset by the VLAN tag size. So for instance if you want to filter `ip proto \tcp` on an interface receiving only VLAN tagged packets then you must use the following filter instead:

```
vlan and (ip proto \tcp)
```

If the network interface receives both tagged and non-tagged packet then this somewhat cumbersome filter must be used:

```
(ip proto \tcp) or (vlan and (ip proto \tcp))
```

10.9 How to do complex searches on domain names?

On search boxes about domain names (Web and DNS reports), you can use a regular expression by prefixing the entry with a *tilde* character (~). For exmaple, you can use this to filter **all but some names**. For instance, here is a valid input to filter all but *Google's* and *Amazon's*:

```
~^(?!(.*\.)?google\.[fr|com]$)(?!.*amazon(\..{2,3}){1,2}$)
```

10.10 How comes my VM keeps losing sync?

Even if you configure NTP on a virtual appliance, ESX "helper" programs will try to set the date and time of the VM from the ESX guest. This process will run concurrently with NTP date synchronisation with undefined results. So if you have a VM that's regularly out of sync make sure your ESX itself has the correct time.

10.11 What about Open Source?

SecurActive uses internationally proven and rock solid open source components such as *Linux*, *Python*, *Zope*, *Postgresql*, *Git*, *GCC* ... Our company has chosen to actively contribute to the open source community by regularly submitting patches to these projects and provide access to parts of its own code [1].

10.12 Standard TCP Session

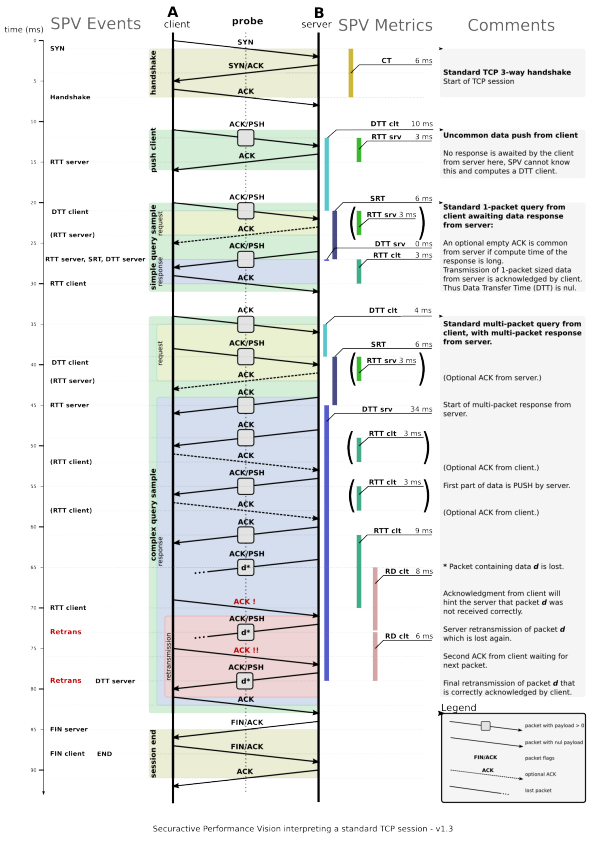


Figure 10.2: Standard TCP Session

[1] https://github.com/securactive

CHAPTER

ELEVEN

KNOWN ISSUES

11.1 Configuration

- If an application is defined with both a webpattern and a client or server zone then all conversations matching this webpattern but not the zone will belongs to NC TCP, even if it should belongs to another application according to the TCP ports.

Note: This webpattern issue is fixed since release 2.10.

11.2 Interface

- There is no error message when a login attempt fails.
- Sometimes where plotting data for the last hour, the chart ends with a zero value.
- In the charts, when a high value is immediately followed by a zero value then the smooth interpolation algorithm makes it go underneath the 0 line just after the 0 value.

Note: This chart issue is fixed since release 2.10.

- When another language than English is requested some buttons are labelled in English nonetheless.

11.3 Various

- SMTP delivery of reports lack retries.
- There is no procedure to delete oldest data whenever the data disks become full.
- Configuration dump/restore don't work across version boundary.

Note: Since release 2.11, you cannot restore a incompatible configuration anymore.

11.4 Sniffer

- In case of IP fragmentation, the timestamps of involved packets are set to the last received one.

11.5 Upgrading

- In some cases, the sniffer may fail to restart after an upgrade and leave some stalled processes if it is restarted on its own with *Pulsar*. One of the possible symptoms is that the `poller` command in Pulsar fails to display the poller and license status. Rebooting solves this issue.

11.6 Metrics

- In versions **prior to 2.9**, the retransmission rate (`RR`) was computed as the number of retransmitted TCP segments divided by the total number of TCP segments. As of version 2.9, it is instead divided by the number of packets liable be retransmitted, such as the TCP segments carrying a payload.
- In versions **prior to 2.9**, keep-alive packets occurring after the completion of a data transfer were taken into account in the computation of the Data Transfert Time (`DTT`) metric, resulting in abnormally large values. In order to avoid this issue, as of version 2.9, data transfers are considered complete after a 1 second timeout.

11.7 Pulsar

- When resizing the datadisk, console display several occurrences of this error message: *parted: sending ioctl XXXX to a partition!*. This can safely been ignored.

CHAPTER

TWELVE

GLOSSARY

Aggregation period Time period over which all data are aggregated into flows (for each set of client, server and application). The Aggregation Period is defined for an aggregation level as time interval over which all flows are aggregated in the database on their IP src/dst, Zone src/dst and Application. Individual flows within the aggregated data cannot be viewed separately; The Aggregation Period defines the data resolution for an aggregation level.

Application Group logical or business related flow to emphase valuable perspective an Application is identified with a name and a color, and defined by a set of Signature or a set of Port Range (at least one non-empty set of either), a set of client and server zones. A conversation is attributed to an application with the following rule: (PORT_RANGE1 OR .... OR PORT_RANGEn OR SIGNATURE1 OR ... OR SIGNATUREn) AND ((SERVER_ZONE1 OR ... OR SERVER_ZONEn) AND (CLIENT_ZONE1 OR ... OR CLIENT_ZONEn)); in case a conversation matches previous rule of several application, the priority will be given to the application whose definition is the most precise, i.e. the thinest port range, signature or server/client zone.

Application NC NC stands for Non Classified. A NC Application is a special application that will match conversations that do not match any configured application.

Application Port Range Port or range of ports. If not used in conjonction with an IP protocol then apply to both 'TCP' or 'UDP'.

Collector Central database and Web GUI of Performance Vision. The collector can also host a local poller, and usually collects statistics from remote pollers.

Connection Time (CT) Time taken by the exchange of the 3-way TCP handshake. CT stands for Connection Time. CT is defined as the duration of the three way handshake (SYN, SYN/ACK, ACK) of TCP session.

Conversation Regroups network exchanges between two network addresses for one application during the observation period. A conversation is defined as a group of flows between a client and a server over an observation period.

Data Transfer Time (DTT) Time spent by the client or the server to send data. The DTT stands for Data Transfer Time. DTT server is defined as the time between the first data packet (with ACK flag and a non null payload) from the server and the last packet considered as part of the same answer. DTT client is the symmetric metric in opposite direction. Packets are considered part of the same answer if packet share the same acknowledgment number ; FIN, RST from server or client. A Timeout will cancel a DTT. Note that if the answer is small enough to be contained in only one packet, the DTT will be of '0'.

Delta sessions Number of session established minus those closed. Delta Session is a metric defined as the difference of the number of opened session to the number of closed session. Negative value means that more session were closed than opened.

Device Identifier Identifies the physical network adapter that received the network traffic associated to a conversation.

DiffServ Code Point (DSCP) 6 bits value taken from the TOS field of IP header, used in some networks to govern the QoS of each packet. No standard meaning being assigned to given values, only the raw numeric value is reported. For a given conversation, the probe keeps only the biggest DSCP encountered. In practice a whole conversation should be governed by a single DSCP value.

End User Response Time Total time the user waited to get an applicative answer. The EURT stands for End User Response Time. EURT is defined as the sum of the RTT (client + server), the SRT and the DTT (client + server). A timeout will cancel the computation of EURT.

Fallback By convention, a zone which collect a larger set of addresses that includes addresses of other, more specific zones. See *Fallbacks* for more details.

Flow Regroups data exchanges between two network addresses for one application on the aggregation period. A flow is a group of communications between two network addresses for one application during the aggregation period. Notice that the VLAN tag, if present, as well as the device identifier, are considered components of the network address.

HTTP hit A HTTP hit designate a single HTTP transaction, used to build a HTTP page. This is typically an image, a script, a stylesheet... The transaction to obtain the HTML is itself considered as a hit. Thus, a page that contains 2 images and 1 stylesheet, all stored in different URLs, is made of 4 hits.

HTTP page A HTTP page is the set of HTTP transactions that was required to build a whole HTML page, including its images, scripts, stylesheets and other related resources. This might also include the resources used to update the page dynamically (through AJAX, that is, Javascript, or from other means.)

Initial Sequence Number The sequence number used in the SYN packet of a TCP connection.

Jitter Packet delay variation. The Jitter is defined as the variance of RTT (average difference between RTT measures and the average RTT). For more details, this equation is used: Sqrt((Average(RTT1**2, ... , RTTn**2) - Average(RTT1, ..., RTTn)**2).

Maximum Transfert Unit (MTU) The MTU that's reported by the probe is the size of the biggest Ethernet frame that was seen in this conversation. It is thus distinct from the physical MTU, although for a large number of packets the observed MTU is expected to converge toward the physical MTU.

Media Access Control (MAC) address Identifier assigned to each network adapter and used for addressing in the lowest physical layer. As in practice only Ethernet devices are supported then these will always be Ethernet addresses.

Observation period In all reports, defines the observation time window. Observation Period is based on a starting time and an ending time provided by the user. These user-defined boundaries will automatically be moved to the closest previous aggregation boundary for the starting time and to the next aggregation boundary for the ending time: this modified time interval is the actual observation period.

Operating System (OS) Different operating systems implement the core network protocols differently. The probe attempts to guess the operating system that's used at both ends of a conversation based on these differences. This information, even if inaccurate and unreliable, can still be used to help identify a host or a network trouble.

Poller Remote probe that listen and analyze the network traffic to produce statistics that the collector will fetch and insert into the central database.

Protocol Stack The various protocols identified in a flow. For instance, an HTTP conversation that's carried in TCP over IP over Ethernet may be reported with the `Eth/IPv4/TCP/HTTP` protocol stack, whereas in case of a IPv6 in v4 tunnel the protocol stack would be `Eth/IPv4/IPv6/TCP/HTTP`. Not only this notation makes all sort of tunnels visible but it also make apparent some protocols that are detected by the sniffer despite running on non standard ports.

Retransmission Packets being resent, when they have either been lost or damaged. Packet Retransmission is identified thanks to their TCP sequence and acknowledgment numbers, and checksum values. Only packets with a non-null payload are checked.

Retransmission Delay (RD) Delay between a packet and it's next retransmission. RD stands for Retransmission Delay. RD is defined as the time between a packet and its next retransmission.

Retransmission Duplicate ACK Duplicate acknowledgment Packet with null payload. Duplicate ACK are TCP ACK packets that are identified thanks to their same acknowledgment value and their empty payload.

Retransmission Rate (RR) RR stands for Retransmission Rate. RR is defined as the ratio of retransmitted packets to the total number of packet with a non-zero payload in a conversation.

Retransmission Total Delay between a packet and the last retransmission. TRD stands for Total Retransmission Delay. TRD is defined as the time between a packet and its last retransmission.

Round Trip Time (RTT) Time between an applicative query and a response at the network level. RTT stands for Round Trip Time. RTT is defined as the time between a packet with a non null payload and the corresponding acknowledgment (a packet with a null payload and the TCP ACK flag).

Server Response Time (SRT) Time between a query and an answer at the applicative level. Server Response Time is the elapsed time between a client packet with a non null payload and the corresponding server response (a packet with a non null payload which number of acknowledgment correspond to the first packet).

Session An established communication channel between two devices using TCP. a Session is defined as TCP communication between 2 devices beginning by a successful Handshake, and ending by a Timeout, or Packet with the RST flag from any of the devices, or a Packet with FIN from any of the device that is acknowledged by a FIN/ACK by the other device and followed by a FIN of this same last device. (no FIN/ACK is necessary to conclude that the connection is closed).

Subnet Set of network addresses that have a common declared IP address routing prefix. A Subnet is defined by an IP address and a netmask.

TCP Handshake 3-Way negociation that is part of TCP for establishing a TCP session. A TCP Handshake is defined between 2 devices as exchange of 3 TCP packets flagged SYN, SYN/ACK, ACK.

Timeout Session end by inactivity. Session Timeout will be reported after 120 seconds of complete inactivity (i.e. no packets seen).

Web Application Pattern Mean of recognizing an Application based on a pattern in the payload. Currently these patterns are checked against HTTP URLs only. The pattern syntax allows hostname and optionally a path separated by '/' (ie: 'www.example.com/my/path', or 'www.example.com'). Notice that a wildcards character * is allowed in domain or path part of the pattern. Only Conversation which are detected to be based on HTTP will have URL of their GET/POST/CONNECT request matched against Web application signature's pattern. A match occurs when the pattern match the complete target URL.

Zone A zone corresponds to the location of a sender or emitter. See *Zones* for more details.

CHAPTER

THIRTEEN

APPENDIX

13.1 Integration with other Tools

13.1.1 How to integrate your APS with Cacti

Importing PV template in your Cacti

We recommend following these easy steps in order to ease the integration of PV hosts into the open source Cacti network monitoring system.

Download configuration files

First, you must download a set of files from our web site [1]. Unzip this archive somewhere before proceeding.

These files are two-fold: three of them, being host templates, can be uploaded from Cacti GUI. The others, begin SNMP query templates, must be copied in Cacti's resource directory.

Upload host templates

Log in your Cacti GUI with the admin user and go to the *Import Templates* page. There you can upload the file named `cacti_host_template_sniffer.xml`:

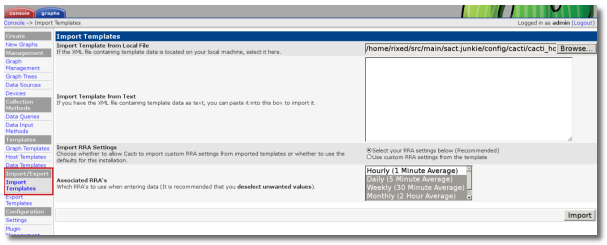


Figure 13.1: Import/Export

Which should bring you to a success page such as:

Repeat this operation for the two other host templates `cacti_host_template_pv_central.xml` and `cacti_host_template_pv_probe.xml`.

Copy data queries

This step is not enough, though, since these templates use custom SNMP data queries. If you go to *Data Queries* you will notice a set of new entries:

[1] http://download.securactive.net/pv/misc/nagios/cacti-config.zip

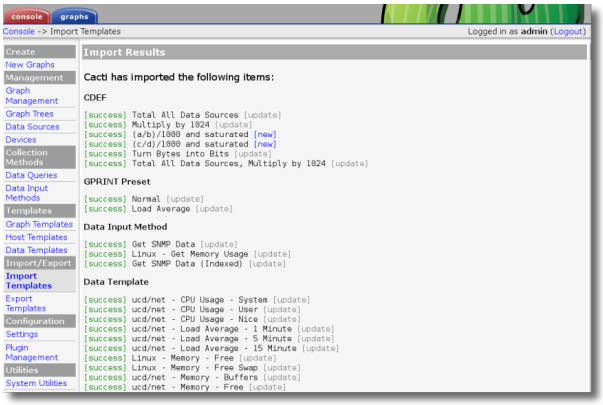

console
graphs
Console -> Import Templates
Logged in as admin (Logout)
Import Results
Cacti has imported the following items:
CDEF
[success] Total All Data Sources [update]
[success] Multiply by 1024 [update]
[success] (a/b)/1000 and saturated [new]
[success] (c/d)/1000 and saturated [new]
[success] Turn Bytes into Bits [update]
[success] Total All Data Sources, Multiply by 1024 [update]
GPRINT Preset
[success] Normal [update]
[success] Load Average [update]
Data Input Method
[success] Get SNMP Data [update]
[success] Linux - Get Memory Usage [update]
[success] Get SNMP Data (Indexed) [update]
Data Template
[success] ucd/net - CPU Usage - System [update]
[success] ucd/net - CPU Usage - User [update]
[success] ucd/net - CPU Usage - Nice [update]
[success] ucd/net - Load Average - 1 Minute [update]
[success] ucd/net - Load Average - 5 Minute [update]
[success] ucd/net - Load Average - 15 Minute [update]
[success] Linux - Memory - Free [update]
[success] Linux - Memory - Free Swap [update]
[success] ucd/net - Memory - Buffers [update]
[success] ucd/net - Memory - Free [update]

Figure 13.2: Imported Items

- *Junkie - Muxer Stats*
- *Junkie - Parser Stats*
- ...
- *PV - BCN*

If you select one of these you will face an error message such as:

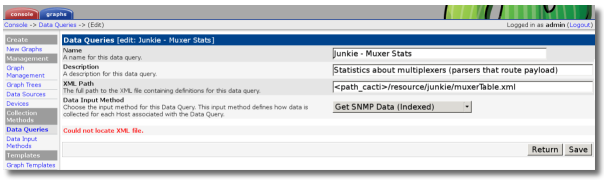


Figure 13.3: Error (XML files not found)

You must manually copy the remaining xml files in the expected path (or change the *XML Path* in each of the data queries definition):

- `muxerTable.xml`, `parserTable.xml` and `sourceTable.xml` in `<path_cacti>/resource/junkie`,
- `bcaTable.xml`, `bcnTable.xml` and `metricTable.xml` in `<path_cacti>/resource/metrics`.

Once you have done this, if you reload any of the *Data Query* you should see the error message replaced by a success indication:

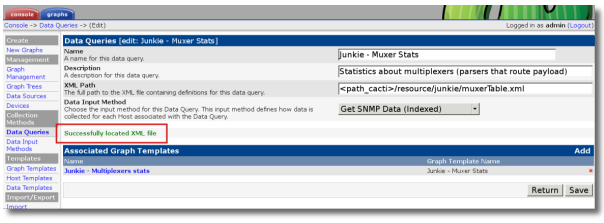


Figure 13.4: Success (XML files found)

You can now proceed with device and graph creation.

Creating a device for your central collector

When choosing a host template select *PV - Central Collector* from the drop down box. Also, take a close look to the SNMP settings for this host. You should choose SNMP version 2, and enter the proper SNMP community, in accordance to the SNMP settings of the collector. You *must* use a bigger SNMP timeout than the default one. On the screenshot below you can see we set a timeout of 20s:

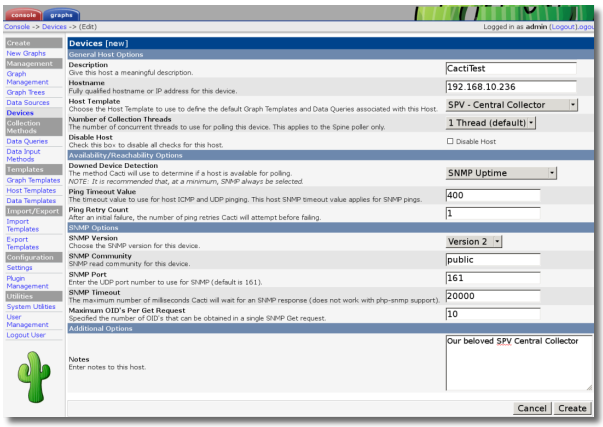


Figure 13.5: Devices Created

Once created, the device will be already populated with a set of data queries appropriate for this kind of host.

Creating graphs for this host

You can now select the *Create Graphs for this Host* link at the top of the device description page. You will be offered plenty graph templates to create, many of which are not relevant for the casual user. We recommend you create one to monitor memory and CPU resource usage:

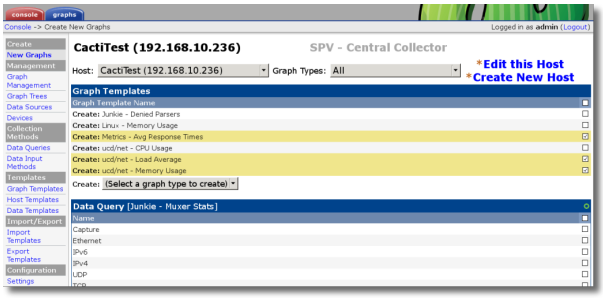


Figure 13.6: Create CPU and memory graphs

As well as one for junkie's packet sources (useful to monitor dups as well as dropped packets) in addition to general purpose interface usage statistics on all listening devices:

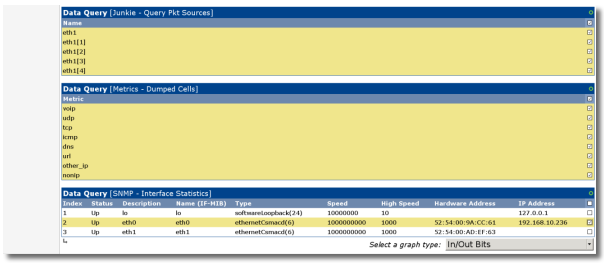

Data Query [Junkie - Query Pkt Sources]
Data Query [Metrics - Dumped Cells]
Data Query [SNMP - Interface Statistics]
Select a graph type:
In/Out Bits

Figure 13.7: Create device / protocol graphs

At the very bottom of the page you will have the opportunity to add graphs to monitor hard disk space (interesting partitions are `/` and `/srv`):

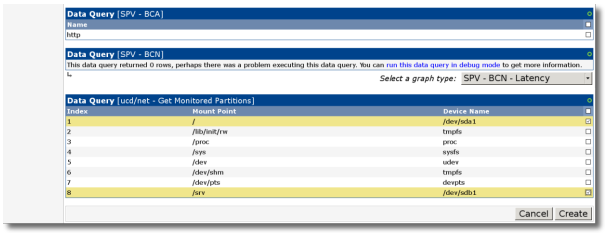

Data Query [SPV - BCA]
Name
http
Data Query [SPV - BCN]
This data query returned 0 rows, perhaps there was a problem executing this data query. You can run this data query in debug mode to get more information.
Select a graph type:
SPV - BCN - Latency
Data Query [ucd/net - Get Monitored Partitions]
Index
Mount Point
Device Name
1 / /dev/sda1
2 /lib/init/rw tmpfs
3 /proc proc
4 /sys sysfs
5 /dev udev
6 /dev/shm tmpfs
7 /dev/pts devpts
8 /srv /dev/sdb1
Cancel
Create

Figure 13.8: Create disk space graphs

If you already have some defined, you will also find a graph template for every BCA and 3 possible graphs for every BCN (for latency, retransmission and traffic):

Figure 13.9: Create BCA and BCN graphs

Once you have checked all graphs you are interested in then click on the *Create* button and if all goes well you should be welcomed with a message such as:

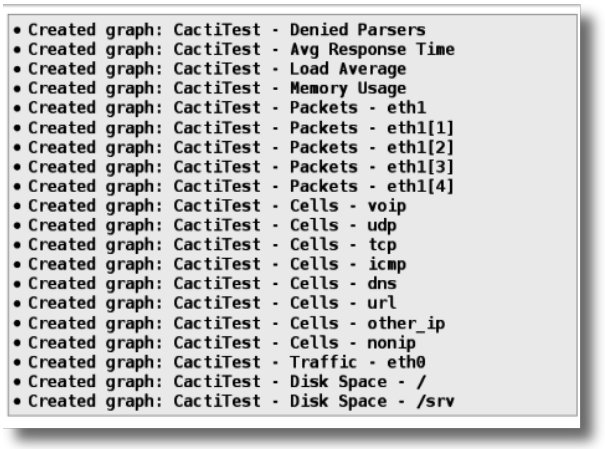

• Created graph: CactiTest - Denied Parsers
• Created graph: CactiTest - Avg Response Time
• Created graph: CactiTest - Load Average
• Created graph: CactiTest - Memory Usage
• Created graph: CactiTest - Packets - eth1
• Created graph: CactiTest - Packets - eth1[1]
• Created graph: CactiTest - Packets - eth1[2]
• Created graph: CactiTest - Packets - eth1[3]
• Created graph: CactiTest - Packets - eth1[4]
• Created graph: CactiTest - Cells - voip
• Created graph: CactiTest - Cells - udp
• Created graph: CactiTest - Cells - tcp
• Created graph: CactiTest - Cells - icmp
• Created graph: CactiTest - Cells - dns
• Created graph: CactiTest - Cells - url
• Created graph: CactiTest - Cells - other_ip
• Created graph: CactiTest - Cells - nonip
• Created graph: CactiTest - Traffic - eth0
• Created graph: CactiTest - Disk Space - /
• Created graph: CactiTest - Disk Space - /srv

Figure 13.10: Summary

Congratulations!

You now have many new graphs to monitor both your PV and your network. Do not forget to create new graphs for the new BCA/BCN you may add in the future.

Creating a device for your probe

Although monitoring a probe is much less interesting since you won't be able to fetch any BCA/BCN from a mere probe, you can still add your PV probes in your Cacti monitoring system by following almost the same steps as above, only selecting a device of type *PV - Probe* instead of *PV - Central Collector*.

Figure 13.11: All graphs displayed

13.1.2 How to integrate your APS with Nagios

Introduction

Nagios is a powerful tool to easily monitor IT network and servers. It can alert on reachability issue, software or hardware problems. But it's much harder to alert on "End User Response Time". With the help of the SNMP module of your PV probes you can add new check on Nagios to perform this and other advanced checks.

Prerequisites

PV side

- The license cannot be a `Free` or `Express` version to enable SNMP
- Enable the PV SNMP via SSH, user admin, with command `snmp start`.
- Create at least one BCA or one BCN.

Network side

- Network access from Nagios server to PV host for protocol SNMP (UDP port 161)

Nagios Side

- A Nagios script can be download from our website (http://download.securactive.net/pv/misc/nagios/nagios_pv_snmp.pl) or on github (https://github.com/securactive/nagios_PV_snmp)
- Nagios v3+
- Perl5+
- Perl libraries : Net::SNMP and Getopt::Long (you can install them with `cpan`)
- Script must be executable by Nagios (`chmod +x nagios_pv_snmp.pl`)

Command-line usage

Help

```
./nagios_pv_snmp.pl --help
```

```
SNMP Network PV Monitor for Nagios version 0.4
GPL licence

Usage: ./nagios_pv_snmp.pl [-h] [-v] -H <host> [-C <snmp_community>] [-2] | (-l login -x passwd [-
-v, --verboes
   print extra debugging information
-h, --help
   print this help message
-H, --hostname=HOST
   name or IP address of host to check
-C, --community=COMMUNITY NAME
   community name for the host's SNMP agent
   default public
-l, --login=LOGIN ; -x, --passwd=PASSWD, -2, --v2c
   Login and auth password for snmpv3 authentication
   If no priv password exists, implies AuthNoPriv
-2, use snmp v2c
-X, --privpass=PASSWD
   Priv password for snmpv3 (AuthPriv protocol)
-L, --protocols=<authproto>,<privproto>
   <authproto> : Authentication protocol (md5|sha : default md5)
   <privproto> : Priv protocole (des|aes : default des)
-P, --port=PORT
   SNMP port (Default 161)
-i, --insensitive
       Case insensitive for regex match
-s, --showall
       Print all, instead of only matching pattern
--onlyfaulty
       Print only faulty services : not OK BCA or BCN
-a, --bca=NAME
   Name of BCA (htpp, ssh ...).
   This is treated as a regexp : -n http will match BCA http, http intranet, https, ...
   If NAME is "", will check all bca
-n, --bcn=NAME
   Name of BCN ("All - /Private/Private_fallback", ...).
   This is treated as a regexp : -n fallback will match all BNC containing "fallback".
   If NAME is "" , will check all bcn
--bcnatob
       Check Only BCN Status A => B
--bcnbtoa
       Check Only BCN Status B => A
-r, --noregexp
   Do not use regexp to match NAME
--landscape
       Print tables in landscape mode, default is portrait mode
-t, --timeout=INTEGER
   timeout for SNMP in seconds (Default: 5)
-V, --version
   prints version number
```

Examples

- check_snmp_securactive.pl -H 10.0.0.1

  check all bca and bcn of probe 10.0.0.1 it will return global status

- check_snmp_securactive.pl -H 10.0.0.1 -C public -2

  check all bca and bcn it will return global status public community (default) and v2c snmp protocol (default)

- check_snmp_securactive.pl -H 10.0.0.1 -a http

check bca, regex match http, case sensitive https => match Http => not match

- check_snmp_securactive.pl -H 10.0.0.1 -a http -i

  check bca, regex match http, case insensitive https => match Http => match

- check_snmp_securactive.pl -H 10.0.0.1 -a http -i -r

  check bca, pattern match http, case insensitive https => not match Http => match

- check_snmp_securactive.pl -H 10.0.0.1 -a "(google|http)"

  check bca, regex match google or http, case sensitive

- check_snmp_securactive.pl -H 10.0.0.1 -a ^http$

  check bca, regex match ^http$, case sensitive http => match https => not match Http => not match

- check_snmp_securactive.pl -H 10.0.0.1 -n

  check all bcn

- check_snmp_securactive.pl -H 10.0.0.1 -n "fallback"

  check bcn, regex math "fallback", case sensitive

- check_snmp_securactive.pl -H 10.0.0.1 -a "^http$" -i -n "internet" –atob

  check bca, regex math "^http$", case insensitive check bcn, regex math "internet", case insensitive, and check only zone "a" to zone "b" way

Nagios configuration file example

This is only an example of nagios configuration file.

```
define command{
        command_name check_securactive
        command_line $USER2$/check_snmp_securactive.pl -H $HOSTADDRESS$
}
define command{
        command_name check_securactive_bca
        command_line $USER2$/check_snmp_securactive.pl -H $HOSTADDRESS$ -a $ARG1$ $ARG2$
}

define command{
        command_name check_securactive_bcn
        command_line $USER2$/check_snmp_securactive.pl -H $HOSTADDRESS$ -n $ARG1$ $ARG2$
}
~
define host {
        host_name   beta
        alias       beta
        address     192.168.30.30
        use         generic-host
}
define service{
       name                    bca_ssh
       service_description     BCA_SSH
       use                     generic-service
       check_command           check_securactive_bca!ssh
       host_name               beta
}
define service{
       name                    bca_pop3s
       service_description     BCA_POP3S
       use                     generic-service
       check_command           check_securactive_bca!pop3s
```

```
        host_name                       beta
}
define service{
        name                            bca_http
        service_description             BCA_http
        use                             generic-service
        check_command                   check_securactive_bca!"http$"
        host_name                       beta
}
[...]
define service{
        name                            bcn_RetDVersInternet
        service_description             BCN_RetDVersInternet
        use                             generic-service
        check_command                   check_securactive_bcn!R.D
        host_name                       beta
}
}
define service{
        name                            bcn_voip
        service_description             BCN_VOIP
        use                             generic-service
        check_command                   check_securactive_bcn!Voip
        host_name                       beta
}
[...]
```

PV BCA/BCN and Nagios BCA/BCN

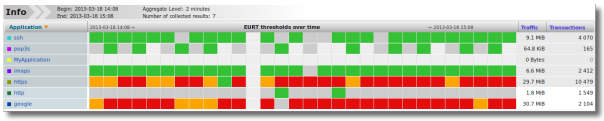


Figure 13.12: BCA on PerformaceVision

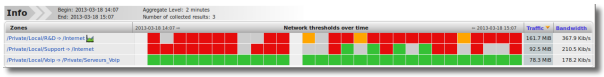
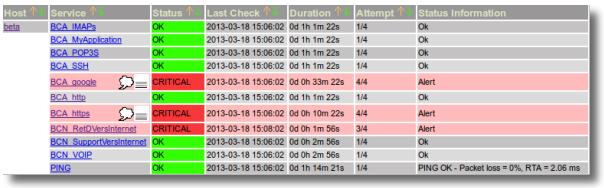


Figure 13.13: BCN on PerformanceVision



Host	Service	Status	Last Check	Duration	Attempt	Status Information
beta	BCA_IMAPs	OK	2013-03-18 15:06:02	0d 1h 1m 22s	1/4	Ok
	BCA_MyApplication	OK	2013-03-18 15:06:02	0d 1h 1m 22s	1/4	Ok
	BCA_POP3S	OK	2013-03-18 15:06:02	0d 1h 1m 22s	1/4	Ok
	BCA_SSH	OK	2013-03-18 15:06:02	0d 1h 1m 22s	1/4	Ok
	BCA_google	CRITICAL	2013-03-18 15:06:02	0d 0h 33m 22s	4/4	Alert
	BCA_http	OK	2013-03-18 15:06:02	0d 1h 1m 22s	1/4	Ok
	BCA_https	CRITICAL	2013-03-18 15:06:02	0d 0h 10m 22s	4/4	Alert
	BCN_RetDVersInternet	CRITICAL	2013-03-18 15:08:02	0d 0h 1m 56s	3/4	Alert
	BCN_SupportVersInternet	OK	2013-03-18 15:06:02	0d 0h 2m 56s	1/4	Ok
	BCN_VOIP	OK	2013-03-18 15:06:02	0d 0h 2m 56s	1/4	Ok
	PING	OK	2013-03-18 15:08:02	0d 1h 14m 21s	1/4	PING OK - Packet loss = 0%, RTA = 2.06 ms

Figure 13.14: BCA and BCN check on Nagios

13.2 Custom Filters

13.2.1 Client/Server

Keyword	Field	Operand Type
0win	Zero Window Size in both direction	*Decimal or hexa.*
0win.count.clt	Zero Window Size from client	*Decimal or hexa.*
0win.count.srv	Zero Window Size from server	*Decimal or hexa.*
app	–	*Application name*
bandw	Total traffic	*Byte quantity*
bandw.clt	Traffic from client to server	*Byte quantity*
bandw.srv	Traffic from server to client	*Byte quantity*
begin	Number of SYN packets	*Decimal or hexa.*
capture.begin	Capture begin time	*Date and time*
capture.end	Capture end time	*Date and time*
ct	Connection time	*Duration*
ct.count	Number of successful handshakes	*Decimal or hexa.*
delta_session	Difference between created session and finished sessions	*Decimal or hexa.*
device	–	*Decimal or hexa.*
diffserv	Client or Server Diffserv	-
diffserv.clt	Client Diffserv	-
diffserv.srv	Server Diffserv	-
dtt	Sum of both DTT client and server	*Duration*
dtt.clt	Data transfer time from client	*Duration*
dtt.count	–	*Decimal or hexa.*
dtt.count.clt	–	*Decimal or hexa.*
dtt.count.srv	–	*Decimal or hexa.*
dtt.srv	Data transfer time from server	*Duration*
dup_ack.count	Total duplicate acks	*Decimal or hexa.*
dup_ack.count.clt	Duplicate acks from client to server	*Decimal or hexa.*
dup_ack.count.srv	Duplicate acks from server to client	*Decimal or hexa.*
end	Number of session finished	*Decimal or hexa.*
eth.proto	Ethernet Type Protocol	*Ethernet Type*
eurt	End-user response time	*Duration*
fin.count	Total number of FIN packets	*Decimal or hexa.*
fin.count.clt	Number of FIN sent by client IP	*Decimal or hexa.*
fin.count.srv	Number FIN sent by server IP	*Decimal or hexa.*
ip	Either client or server IP or subnet	*Address or netmask*
ip.clt	IP which demand a connection to a server	*Address or netmask*
ip.netflow	IP of the netflow capture	*Address or netmask*
ip.srv	IP which replied to a connection demand	*Address or netmask*
mac	Client or Server MAC address	*MAC address*
mac.clt	Client MAC (physical) address	*MAC address*
mac.srv	Server MAC (physical) address	*MAC address*
mtu.clt	Client MTU (Maximum Transmission Unit)	*Decimal or hexa.*
mtu.srv	Server MTU (Maximum Transmission Unit)	*Decimal or hexa.*
os	Client or Server Operating System	*OS name*
os.clt	Client Operating System	*OS name*
os.srv	Server Operating System	*OS name*
payload	Total payload	*Byte quantity*
payload.clt	Payload from client to server	*Byte quantity*
payload.count	Number of IP packets with a payload	*Decimal or hexa.*
payload.count.clt	Number of packets with payload sent from client	*Decimal or hexa.*
payload.count.srv	Number of packets with a payload sent from server	*Decimal or hexa.*
payload.ret	Total retransmission payload	*Byte quantity*
payload.ret.clt	Retransmission payload from client to server	*Byte quantity*
payload.ret.srv	Retransmission payload from server to client	*Byte quantity*
payload.srv	Payload from server to client	*Byte quantity*
pkt.count	Number of IP packets	*Decimal or hexa.*
pkt.count.clt	Number of packets sent from client	*Decimal or hexa.*
pkt.count.srv	Number of packets sent from server	*Decimal or hexa.*

Continued on next page

Table 13.1 – continued from previous page

Keyword	Field	Operand Type
poller.name	Poller name (distributed probe)	*String*
port.srv	Server Port	*Port number*
proto	Protocol	-
protostack	Protocols stack	*Wildcard or regex*
rd	Retransmission delay	*Duration*
rd.clt	Retransmission delay from client to server	*Duration*
rd.count	Retransmission count (both directions)	*Decimal or hexa.*
rd.indic	Total retransmission delay indic	*Duration*
rd.indic.clt	Retransmission delay indic client to server	*Duration*
rd.indic.srv	Retransmission delay indic server to client	*Duration*
rd.rate	Total retransmission rate	*Rate*
rd.rate.clt	Retransmission rate client to server	*Rate*
rd.rate.srv	Retransmission rate server to client	*Rate*
rd.srv	Retransmission delay from server to client	*Duration*
ret	Total retransmission traffic	*Byte quantity*
ret.clt	Retransmission traffic from client to server	*Byte quantity*
ret.srv	Retransmission traffic from server to client	*Byte quantity*
rst.count	Total Number of RST sent	*Decimal or hexa.*
rst.count.clt	Number of RST sent by client IP	*Decimal or hexa.*
rst.count.srv	Number of RST sent by server IP	*Decimal or hexa.*
rtt	Sum of RTTin both directions	*Duration*
rtt.clt	RTT for data from server to client	*Duration*
rtt.count.clt	Number of RTT for data from server to client	*Decimal or hexa.*
rtt.count.srv	Number of RTT for data from client to server	*Decimal or hexa.*
rtt.srv	RTT for data from client to server	*Duration*
srt	Server response time	*Duration*
srt.count	Number of SRT computed in a time interval	*Decimal or hexa.*
timeout	Number of timeout sent	*Decimal or hexa.*
vlan	Tagged Link (802.1Q)	*Decimal or hexa.*
voip.traffic	Total traffic in both directions	*Byte quantity*
zone	Server or Client Zone	*Zone name*
zone.clt	Zone of the client IP	*Zone name*
zone.srv	Zone of the server IP	*Zone name*

13.2.2 Source/Destination

Keyword	Field	Operand Type
0win	Zero Window Size in both direction	*Decimal or hexa.*
0win.count.clt	Zero Window Size from client	*Decimal or hexa.*
0win.count.srv	Zero Window Size from server	*Decimal or hexa.*
app	–	*Application name*
bandw	Total traffic from source to destination	*Byte quantity*
bandw.clt	Total traffic from source to destination	*Byte quantity*
bandw.srv	Traffic from server to client	*Byte quantity*
begin	Number of SYN packets	*Decimal or hexa.*
capture.begin	Capture begin time	*Date and time*
capture.end	Capture end time	*Date and time*
ct	Connection time	*Duration*
ct.count	Number of successful handshakes	*Decimal or hexa.*
delta_session	Difference between created session and finished sessions	*Decimal or hexa.*
device	–	*Decimal or hexa.*
diffserv	Client or Server Diffserv	-
diffserv.clt	Client Diffserv	-
diffserv.srv	Server Diffserv	-

Continued on next page

Table 13.2 – continued from previous page

Keyword	Field	Operand Type
dtt	Oriented DTT	*Duration*
dtt.clt	Data transfer time from client	*Duration*
dtt.count	–	*Decimal or hexa.*
dtt.count.clt	–	*Decimal or hexa.*
dtt.count.srv	–	*Decimal or hexa.*
dtt.srv	Data transfer time from server	*Duration*
dup_ack.count	Total duplicate acks	*Decimal or hexa.*
dup_ack.count.clt	Duplicate acks from client to server	*Decimal or hexa.*
dup_ack.count.srv	Duplicate acks from server to client	*Decimal or hexa.*
end	Number of session finished	*Decimal or hexa.*
eth.proto	Ethernet Type Protocol	*Ethernet Type*
eurt	End-user response time	*Duration*
fin.count	Total number of FIN packets	*Decimal or hexa.*
fin.count.clt	Number of FIN sent by client IP	*Decimal or hexa.*
fin.count.srv	Number FIN sent by server IP	*Decimal or hexa.*
ip	Either source or destination IP or subnet	*Address or netmask*
ip.dst	IP address to which network communication is sent	*Address or netmask*
ip.netflow	IP of the netflow capture	*Address or netmask*
ip.src	IP address from which network communication originates	*Address or netmask*
mac	Client or server MAC address	*MAC address*
mac.dst	Destination MAC Address	*MAC address*
mac.src	Source MAC Address	*MAC address*
mtu	Oriented Max Tranfert Unit	*Decimal or hexa.*
mtu.clt	Client MTU (Maximum Transmission Unit)	*Decimal or hexa.*
mtu.srv	Server MTU (Maximum Transmission Unit)	*Decimal or hexa.*
os	Source or Destination OS	*OS name*
os.clt	Source OS	*OS name*
os.srv	Destination OS	*OS name*
payload	Total payload from source to destination	*Byte quantity*
payload.clt	Total payload from source to destination	*Byte quantity*
payload.count	Number of IP packets with a payload	*Decimal or hexa.*
payload.count.clt	Number of packets with payload sent from client	*Decimal or hexa.*
payload.count.srv	Number of packets with a payload sent from server	*Decimal or hexa.*
payload.ret	Total retransmission payload	*Byte quantity*
payload.ret.clt	Retransmission payload from client to server	*Byte quantity*
payload.ret.srv	Retransmission payload from server to client	*Byte quantity*
payload.srv	Payload from server to client	*Byte quantity*
pkt.count	Total number of IP packets	*Decimal or hexa.*
pkt.count.clt	Number of packets sent from client	*Decimal or hexa.*
pkt.count.srv	Number of packets sent from server	*Decimal or hexa.*
poller.name	Poller name (distributed probe)	*String*
port.srv	Server Port	*Port number*
proto	Protocol	-
protostack	Protocols stack	*Wildcard or regex*
rd	Retransmission delay	*Duration*
rd.clt	Retransmission delay from client to server	*Duration*
rd.count	Retransmission count (both directions)	*Decimal or hexa.*
rd.indic	Total retransmission delay indic	*Duration*
rd.indic.clt	Retransmission delay indic client to server	*Duration*
rd.indic.srv	Retransmission delay indic server to client	*Duration*
rd.rate	Oriented retransmission rate	*Rate*
rd.rate.clt	Retransmission rate client to server	*Rate*
rd.rate.srv	Retransmission rate server to client	*Rate*
rd.srv	Retransmission delay from server to client	*Duration*
ret	Total retransmission traffic	*Byte quantity*

Continued on next page

Table 13.2 – continued from previous page

Keyword	Field	Operand Type
ret.clt	Retransmission traffic from client to server	*Byte quantity*
ret.srv	Retransmission traffic from server to client	*Byte quantity*
rst.count	Total Number of RST sent	*Decimal or hexa.*
rst.count.clt	Number of RST sent by client IP	*Decimal or hexa.*
rst.count.srv	Number of RST sent by server IP	*Decimal or hexa.*
rtt	Oriented RTT	*Duration*
rtt.clt	RTT for data from server to client	*Duration*
rtt.count	–	*Decimal or hexa.*
rtt.count.clt	Number of RTT for data from server to client	*Decimal or hexa.*
rtt.count.srv	Number of RTT for data from client to server	*Decimal or hexa.*
rtt.srv	RTT for data from client to server	*Duration*
srt	Server response time	*Duration*
srt.count	Number of SRT computed in a time interval	*Decimal or hexa.*
timeout	Number of timeout sent	*Decimal or hexa.*
vlan	Tagged Link (802.1Q)	*Decimal or hexa.*
voip.traffic	Total traffic in both directions	*Byte quantity*
zone	Source or Destination Zone	*Zone name*
zone.dst	Zone name to which network communication is sent	*Zone name*
zone.src	Zone name from which network communication originates	*Zone name*

13.2.3 HTTP

Keyword	Field	Operand Type
capture.begin	Capture begin time	*Date and time*
capture.end	Capture end time	*Date and time*
device	–	*Decimal or hexa.*
http.clt.dtt	Average query transfert time.	*Duration*
http.hit.count	Sum of HTTP hits	*Decimal or hexa.*
http.hit.err.count	Sum of Hits with an error status (4xx and 5xx).	*Decimal or hexa.*
http.hit.rt	Average of the hit response time	*Duration*
http.host	URL Host	*String*
http.page.count	Sum of HTTP pages	*Decimal or hexa.*
http.page.lt	Average of page load time	*Duration*
http.request.length	Sum of content length generated by HTTP queries.	*Byte quantity*
http.request.method	The HTTP method used to query.	*HTTP Method*
http.response.dtt	Average response transfert time.	*Duration*
http.response.length	Sum of content length generated by HTTP responses.	*Byte quantity*
http.response.server	Software declared as the HTTP server	*String*
http.response.status	The HTTP response code (1xx to 5xx)	*HTTP status*
http.url.path	URL Path	*Wildcard or regex*
http.user_agent	User agent	*String*
ip	Either client or server IP or subnet	*Address or netmask*
ip.clt	IP which demand a connection to a server	*Address or netmask*
ip.srv	IP which replied to a connection demand	*Address or netmask*
mac	Client or Server MAC address	*MAC address*
mac.clt	Client MAC (physical) address	*MAC address*
mac.srv	Server MAC (physical) address	*MAC address*
pkt.count	Number of IP packets	*Decimal or hexa.*
poller.name	Poller name (distributed probe)	*String*
port.srv	Server Port	*Port number*
vlan	Tagged Link (802.1Q)	*Decimal or hexa.*
zone	Server or Client Zone	*Zone name*
zone.clt	Zone of the client IP	*Zone name*
zone.srv	Zone of the server IP	*Zone name*

13.2.4 SQL

Keyword	Field	Operand Type
capture.begin	Capture begin time	*Date and time*
capture.end	Capture end time	*Date and time*
device	–	*Decimal or hexa.*
ip	Either client or server IP or subnet	*Address or netmask*
ip.clt	IP which demand a connection to a server	*Address or netmask*
ip.srv	IP which replied to a connection demand	*Address or netmask*
mac	Client or Server MAC address	*MAC address*
mac.clt	Client MAC (physical) address	*MAC address*
mac.srv	Server MAC (physical) address	*MAC address*
poller.name	Poller name (distributed probe)	*String*
protostack	Protocols stack	*Wildcard or regex*
sql.clt.dtt	Average query transfert time.	*Duration*
sql.dbname	The database or instance name which is used to execute the...	*Wildcard or regex*
sql.dbuser	Authenticated username who execute the queries	*Wildcard or regex*
sql.error.code	The system specific error code	*String*
sql.error.count	Number of errors	*Decimal or hexa.*
sql.error.msg	The SQL error message	*String*
sql.error.rate	Errors rate	*Rate*
sql.error.status	The SQL error status	*String*
sql.query.command	Type of SQL command	*SQL command*
sql.query.count	Number of queries	*Decimal or hexa.*
sql.query.packets	Query packets at applicative level (PDU)	*Decimal or hexa.*
sql.query.payload	Sum of query payload	*Byte quantity*
sql.response.dtt	Average response transfert time.	*Duration*
sql.response.packets	Response packets at applicative level (PDU)	*Decimal or hexa.*
sql.response.payload	Sum of response payload	*Byte quantity*
sql.system	Database system	*SQL system*
srt	Server response time	*Duration*
srt.count	Number of SRT computed in a time interval	*Decimal or hexa.*
vlan	Tagged Link (802.1Q)	*Decimal or hexa.*
zone	Server or Client Zone	*Zone name*
zone.clt	Zone of the client IP	*Zone name*
zone.srv	Zone of the server IP	*Zone name*

13.2.5 CIFS

Keyword	Field	Operand Type
capture.begin	Capture begin time	*Date and time*
capture.end	Capture end time	*Date and time*
cifs.command	CIFS Command	*SMB command*
cifs.data.payload	Payload of data (files transfered) without CIFS meta infor...	*Byte quantity*
cifs.domain	CIFS Domain	*String*
cifs.error.count	Number of errors, mostly server side.	*Decimal or hexa.*
cifs.fileid	CIFS File ID	*Decimal or hexa.*
cifs.meta.payload	Metadata payload used for the CIFS commands, like 'move', ...	*Byte quantity*
cifs.meta.read	Number of metadata bytes read.	*Byte quantity*
cifs.meta.written	Number of metadata bytes written.	*Byte quantity*
cifs.path	CIFS Path to the file related to this command	*Wildcard or regex*
cifs.query.count	Number of queries	*Decimal or hexa.*
cifs.query.packets	Query packets at applicative level (PDU)	*Decimal or hexa.*
cifs.query.payload	Sum of query payload	*Byte quantity*
cifs.query.write	Number of bytes to be written.	*Byte quantity*

Continued on next page

Table 13.5 – continued from previous page

Keyword	Field	Operand Type
cifs.response.packets	Response packets at applicative level (PDU)	*Decimal or hexa.*
cifs.response.payload	Sum of response payload	*Byte quantity*
cifs.response.read	Number of bytes read.	*Byte quantity*
cifs.response.write	Number of bytes effectively written.	*Byte quantity*
cifs.status	CIFS Status	*SMB status*
cifs.subcommand	CIFS Subcommand	*SMB sub-command*
cifs.tree	CIFS Tree related to this command	*Wildcard or regex*
cifs.tree.id	CIFS Tree ID	*Decimal or hexa.*
cifs.user	CIFS User	*String*
cifs.warning.count	Number of warnings, mostly client side.	*Decimal or hexa.*
device	–	*Decimal or hexa.*
ip	Either client or server IP or subnet	*Address or netmask*
ip.clt	IP which demand a connection to a server	*Address or netmask*
ip.srv	IP which replied to a connection demand	*Address or netmask*
mac	Client or Server MAC address	*MAC address*
mac.clt	Client MAC (physical) address	*MAC address*
mac.srv	Server MAC (physical) address	*MAC address*
poller.name	Poller name (distributed probe)	*String*
protostack	Protocols stack	*Wildcard or regex*
srt	Server response time	*Duration*
srt.count	Number of SRT computed in a time interval	*Decimal or hexa.*
vlan	Tagged Link (802.1Q)	*Decimal or hexa.*
zone	Server or Client Zone	*Zone name*
zone.clt	Zone of the client IP	*Zone name*
zone.srv	Zone of the server IP	*Zone name*

13.2.6 ICMP

Keyword	Field	Operand Type
app	–	*Application name*
bandw	Total traffic	*Byte quantity*
bandw.clt	Traffic from client to server	*Byte quantity*
bandw.srv	Traffic from server to client	*Byte quantity*
capture.begin	Capture begin time	*Date and time*
capture.end	Capture end time	*Date and time*
device	–	*Decimal or hexa.*
diffserv	Client or Server Diffserv	-
diffserv.clt	Client Diffserv	-
diffserv.srv	Server Diffserv	-
eth.proto	Ethernet Type Protocol	*Ethernet Type*
icmp.code	ICMP code	*Decimal or hexa.*
icmp.err.ip.clt	Source IP of the ICMP error	*Address or netmask*
icmp.err.ip.srv	Destination IP og the ICMP error	*Address or netmask*
icmp.err.port	ICMP error port	*Port number*
icmp.err.zone.clt	Source zone of the ICMP error	*Zone name*
icmp.err.zone.srv	Destination zone of the ICMP error	*Zone name*
icmp.type	ICMP type	*Decimal or hexa.*
ip	Either client or server IP or subnet	*Address or netmask*
ip.clt	IP which send the packet	*Address or netmask*
ip.netflow	IP of the netflow capture	*Address or netmask*
ip.srv	IP which replied to a connection demand	*Address or netmask*
mac	Client or Server MAC address	*MAC address*
mac.clt	Client MAC (physical) address	*MAC address*
mac.srv	Server MAC (physical) address	*MAC address*

Continued on next page

Table 13.6 – continued from previous page

Keyword	Field	Operand Type
mtu.clt	Client MTU (Maximum Transmission Unit)	*Decimal or hexa.*
mtu.srv	Server MTU (Maximum Transmission Unit)	*Decimal or hexa.*
pkt.count	Number of IP packets	*Decimal or hexa.*
pkt.count.clt	Number of packets sent from client	*Decimal or hexa.*
pkt.count.srv	Number of packets sent from server	*Decimal or hexa.*
poller.name	Poller name (distributed probe)	*String*
proto	Protocol	-
protostack	Protocols stack	*Wildcard or regex*
vlan	Tagged Link (802.1Q)	*Decimal or hexa.*
voip.traffic	Total traffic in both directions	*Byte quantity*
zone	Server or Client Zone	*Zone name*
zone.clt	Zone from the ICMP packet was sent	*Zone name*
zone.srv	Zone of the server IP	*Zone name*

13.2.7 DNS

Keyword	Field	Operand Type
capture.begin	Capture begin time	*Date and time*
capture.end	Capture end time	*Date and time*
device	–	*Decimal or hexa.*
dns.bandw	Total traffic	*Byte quantity*
dns.bandw.clt	Traffic from client to server	*Byte quantity*
dns.bandw.srv	Traffic from server to client	*Byte quantity*
dns.pkt.count	Number of IP packets	*Decimal or hexa.*
dns.pkt.count.clt	Number of packets sent from client	*Decimal or hexa.*
dns.pkt.count.srv	Number of packets sent from server	*Decimal or hexa.*
dns.req.class	The DNS class of the request.	*DNS class*
dns.req.name	The name or IP address to resolve.	*Wildcard or regex*
dns.req.type	The DNS type of the request.	*DNS Type*
dns.res.class	The DNS class of the response.	*DNS class*
dns.res.name	The response to the DNS name resolution request.	*String*
dns.res.rcode	Code of DNS response.	*DNS result*
dns.res.type	The DNS type of the response.	*DNS Type*
drt	DNS response time.	*Duration*
drt.count	Number of DRT computed in a time interval.	*Decimal or hexa.*
eth.proto	Ethernet Type Protocol	*Ethernet Type*
ip	Either client or server IP or subnet	*Address or netmask*
ip.requester	Source IP which issued the DNS request for resolution	*Address or netmask*
ip.srv	IP which replied to a connection demand	*Address or netmask*
mac	Client or Server MAC address	*MAC address*
mac.clt	Client MAC (physical) address	*MAC address*
mac.srv	Server MAC (physical) address	*MAC address*
poller.name	Poller name (distributed probe)	*String*
proto	Protocol	-
vlan	Tagged Link (802.1Q)	*Decimal or hexa.*
voip.traffic	Total traffic in both directions	*Byte quantity*
zone	Server or Client Zone	*Zone name*
zone.clt	Zone from where the DNS request came from	*Zone name*
zone.srv	Zone of the server IP	*Zone name*

13.2.8 Non IP

Keyword	Field	Operand Type
app	–	*Application name*
bandw	Total traffic	*Byte quantity*
bandw.clt	Traffic from client to server	*Byte quantity*
bandw.srv	Traffic from server to client	*Byte quantity*
capture.begin	Capture begin time	*Date and time*
capture.end	Capture end time	*Date and time*
device	–	*Decimal or hexa.*
eth.proto	Ethernet Protocol	*Ethernet Type*
mac	Client or Server MAC address	*MAC address*
mac.clt	Client MAC (physical) address	*MAC address*
mac.srv	Server MAC (physical) address	*MAC address*
mtu.clt	Client MTU (Maximum Transmission Unit)	*Decimal or hexa.*
mtu.srv	Server MTU (Maximum Transmission Unit)	*Decimal or hexa.*
pkt.count	Number of IP packets	*Decimal or hexa.*
pkt.count.clt	Number of packets sent from client	*Decimal or hexa.*
pkt.count.srv	Number of packets sent from server	*Decimal or hexa.*
poller.name	Poller name (distributed probe)	*String*
proto	Protocol	-
protostack	Protocols stack	*Wildcard or regex*
vlan	Tagged Link (802.1Q)	*Decimal or hexa.*
voip.traffic	Total traffic in both directions	*Byte quantity*
zone	Server or Client Zone	*Zone name*
zone.clt	Zone of the client IP	*Zone name*
zone.srv	Zone of the server IP	*Zone name*

13.2.9 VoIP

Keyword	Field	Operand Type
0win	Zero Window Size in both direction	*Decimal or hexa.*
0win.count.clt	Zero Window Size from client	*Decimal or hexa.*
0win.count.srv	Zero Window Size from server	*Decimal or hexa.*
app	–	*Application name*
bandw	Total traffic	*Byte quantity*
bandw.clt	Traffic from caller	*Byte quantity*
bandw.srv	Traffic from callee	*Byte quantity*
begin	Number of SYN packets	*Decimal or hexa.*
capture.begin	Capture begin time	*Date and time*
capture.end	Capture end time	*Date and time*
ct	Connection time	*Duration*
ct.count	Number of successful handshakes	*Decimal or hexa.*
delta_session	Difference between created session and finished sessions	*Decimal or hexa.*
device	–	*Decimal or hexa.*
diffserv	Client or Server Diffserv	-
diffserv.clt	Client Diffserv	-
diffserv.srv	Server Diffserv	-
dtt	Sum of both DTT client and server	*Duration*
dtt.clt	Data transfer time from client	*Duration*
dtt.count	–	*Decimal or hexa.*
dtt.count.clt	–	*Decimal or hexa.*
dtt.count.srv	–	*Decimal or hexa.*
dtt.srv	Data transfer time from server	*Duration*
dup_ack.count	Total duplicate acks	*Decimal or hexa.*

Continued on next page

Table 13.8 – continued from previous page

Keyword	Field	Operand Type
dup_ack.count.clt	Duplicate acks from client to server	*Decimal or hexa.*
dup_ack.count.srv	Duplicate acks from server to client	*Decimal or hexa.*
end	Number of session finished	*Decimal or hexa.*
eth.proto	Ethernet Type Protocol	*Ethernet Type*
eurt	End-user response time	*Duration*
fin.count	Total number of FIN packets	*Decimal or hexa.*
fin.count.clt	Number of FIN sent by client IP	*Decimal or hexa.*
fin.count.srv	Number FIN sent by server IP	*Decimal or hexa.*
ip	Either client or server IP or subnet	*Address or netmask*
ip.clt	IP which demand a connection to a server	*Address or netmask*
ip.netflow	IP of the netflow capture	*Address or netmask*
ip.srv	IP which replied to a connection demand	*Address or netmask*
mac	Client or Server MAC address	*MAC address*
mac.clt	Client MAC (physical) address	*MAC address*
mac.srv	Server MAC (physical) address	*MAC address*
mtu.clt	Client MTU (Maximum Transmission Unit)	*Decimal or hexa.*
mtu.srv	Server MTU (Maximum Transmission Unit)	*Decimal or hexa.*
os	Client or Server Operating System	*OS name*
os.clt	Client Operating System	*OS name*
os.srv	Server Operating System	*OS name*
payload	Total payload	*Byte quantity*
payload.clt	Payload from client to server	*Byte quantity*
payload.count	Number of IP packets with a payload	*Decimal or hexa.*
payload.count.clt	Number of packets with payload sent from client	*Decimal or hexa.*
payload.count.srv	Number of packets with a payload sent from server	*Decimal or hexa.*
payload.ret	Total retransmission payload	*Byte quantity*
payload.ret.clt	Retransmission payload from client to server	*Byte quantity*
payload.ret.srv	Retransmission payload from server to client	*Byte quantity*
payload.srv	Payload from server to client	*Byte quantity*
pkt.count	Number of IP packets	*Decimal or hexa.*
pkt.count.clt	Number of packets sent from client	*Decimal or hexa.*
pkt.count.srv	Number of packets sent from server	*Decimal or hexa.*
poller.name	Poller name (distributed probe)	*String*
port.srv	Server Port	*Port number*
proto	Protocol	-
protostack	Protocols stack	*Wildcard or regex*
rd	Retransmission delay	*Duration*
rd.clt	Retransmission delay from client to server	*Duration*
rd.count	Retransmission count (both directions)	*Decimal or hexa.*
rd.indic	Total retransmission delay indic	*Duration*
rd.indic.clt	Retransmission delay indic client to server	*Duration*
rd.indic.srv	Retransmission delay indic server to client	*Duration*
rd.rate	Total retransmission rate	*Rate*
rd.rate.clt	Retransmission rate client to server	*Rate*
rd.rate.sign	Total retransmission rate for signalization	*Rate*
rd.rate.srv	Retransmission rate server to client	*Rate*
rd.sign	Signalization retransmission delay	*Duration*
rd.sign.count	Number of sign. retransmission (both directions)	*Decimal or hexa.*
rd.srv	Retransmission delay from server to client	*Duration*
ret	Total retransmission traffic	*Byte quantity*
ret.clt	Retransmission traffic from client to server	*Byte quantity*
ret.srv	Retransmission traffic from server to client	*Byte quantity*
rst.count	Total Number of RST sent	*Decimal or hexa.*
rst.count.clt	Number of RST sent by client IP	*Decimal or hexa.*
rst.count.srv	Number of RST sent by server IP	*Decimal or hexa.*

Continued on next page

Table 13.8 – continued from previous page

Keyword	Field	Operand Type
`rtt`	Sum of RTTin both directions	*Duration*
`rtt.clt`	RTT for data from server to client	*Duration*
`rtt.count.clt`	Number of RTT for data from server to client	*Decimal or hexa.*
`rtt.count.srv`	Number of RTT for data from client to server	*Decimal or hexa.*
`rtt.sign`	Sum of signalization RTT in both directions	*Duration*
`rtt.sign.clt`	RTT for signalization data from server to client	*Duration*
`rtt.sign.count.clt`	Number of RTT for signalization data from server to client	*Decimal or hexa.*
`rtt.sign.count.srv`	Number of RTT for signalization data from client to server	*Decimal or hexa.*
`rtt.sign.srv`	RTT for signalization data from client to server	*Duration*
`rtt.srv`	RTT for data from client to server	*Duration*
`srt`	Server response time	*Duration*
`srt.count`	Number of SRT computed in a time interval	*Decimal or hexa.*
`srt.sign`	Server response time for signalization	*Duration*
`srt.sign.count`	Number of signalization transactions in a time interval	*Decimal or hexa.*
`timeout`	Number of timeout sent	*Decimal or hexa.*
`vlan`	Tagged Link (802.1Q)	*Decimal or hexa.*
`voip.traffic`	Voice total traffic	*Byte quantity*
`voip.traffic.sign`	Signalization total traffic	*Byte quantity*
`zone`	Server or Client Zone	*Zone name*
`zone.clt`	Zone of the client IP	*Zone name*
`zone.srv`	Zone of the server IP	*Zone name*

Type definitions

13.2.10 Address or netmask

This can be either a complete IPv4 or IPv6 address, or it can also be an IP address completed with wildcards patterns `*` to form a netmask.

Operators: !=, =

- Example of *valid* inputs: `192.168.*.*`, `192.168.5.10`
- Example of *invalid* inputs: `192.524.1.1`

13.2.11 Application name

This value must be a valid application name, enclosed between quotes.

Operators: !=, =

- Example of *valid* inputs: `http`
- Example of *invalid* inputs: `unknown-app`

13.2.12 Byte quantity

This value indicates a quantity of bytes withits unit. Note that there's no space between the quantity and the unit.

Operators: !=, <, <=, =, >, >=

- Example of *valid* inputs: `42B`, `4KB`, `4KiB`, `56MiB`
- Example of *invalid* inputs: `4 KiB`

13.2.13 DNS Type

A DNS type value, either numeric or symbolic.

Operators: !=, =

- Example of *valid* inputs: `4`, `A`, `MX`
- Example of *invalid* inputs: `1223648`, `FOO`

13.2.14 DNS class

A DNS class, either numeric or symbolic.

Operators: !=, =

- Example of *valid* inputs: `1`, `IN`
- Example of *invalid* inputs: `A`, `MX`

13.2.15 DNS result

A DNS result code, either numeric or symbolic.

Operators: !=, =

- Example of *valid* inputs: `0`, `NoError`, `ServFail`
- Example of *invalid* inputs: `45778`, `SomeCode`

13.2.16 Date and time

A date and time value of the following format : `YYYY-MM-DD hh:mm`. Note that the value must be enclosed between simple or double quotes.

Operators: !=, <, <=, =, >, >=

- Example of *valid* inputs: `"2000-01-01 00:00"`, `'2012-06-14 17:15'`
- Example of *invalid* inputs: `"2000-01-01"`, `2013/11/02 14:58`

13.2.17 Decimal or hexa.

Either a decimal number or an hexadecimal number which must be prefixed by `0x`.

Operators: !=, <, <=, =, >, >=

- Example of *valid* inputs: `0x21`, `0x7a5E`, `4`
- Example of *invalid* inputs: `0X45`, `0xTH`

13.2.18 Duration

A duration in microseconds, minutes, etc. depending on the unit set. The lowest value is in microsecond, specified as `us` or `µs`.

Operators: !=, <, <=, =, >, >=

- Example of *valid* inputs: `42us`, `4µs`, `5m`
- Example of *invalid* inputs: `4 microseconds`

13.2.19 Ethernet Type

The ethernet protocol ID

Operators: !=, =

- Example of *valid* inputs: `"IPv4"`, `0x0800`, `2048`
- Example of *invalid* inputs: `"FOO"`, `123456789`

13.2.20 HTTP Method

A symbol representing the HTTP method name.

Operators: !=, =

- Example of *valid* inputs: `GET`, `HEAD`
- Example of *invalid* inputs: `foo`, `get`

13.2.21 HTTP status

A HTTP status number, or a symbol representing the category of HTTP status number: `Success` will correspond to all HTTP "successful" codes.

Operators: !=, =

- Example of *valid* inputs: `404`, `Success`
- Example of *invalid* inputs: `GET`

13.2.22 MAC address

A MAC address of the form `XX:XX:XX:XX:XX:XX`, where `XX` is a hexadecimal number.

Operators: !=, =

- Example of *valid* inputs: `01:23:45:67:89:ab`, `FF:ab:45:7b:D6:55`
- Example of *invalid* inputs: `AA:AA:AA:AA`

13.2.23 OS name

The name of an operating system, like `'linux'` or `'windows'`. Note that the value must be enclosed between single or double quotes.

Operators: !=, =

- Example of *valid* inputs: `"linux"`, `'windows'`
- Example of *invalid* inputs: `unknown os`

13.2.24 Port number

The value represents a TCP or UDP port number as a numeric value. It can also be given as a port range as in `45-80`.

Operators: !=, <, <=, =, >, >=

- Example of *valid* inputs: `75-110`, `80`
- Example of *invalid* inputs: `85-12`

13.2.25 Protocol name

This value represents the name of a protocol, like `tcp`, or `icmp`.

Operators: !=, =

- Example of *valid* inputs: `icmp`, `mtp`, `tcp`
- Example of *invalid* inputs: `FOO`

13.2.26 Rate

A numeric value as a percentage. The value can be lower than `1%`, as in `0.024%`.

Operators: !=, <, <=, =, >, >=

- Example of *valid* inputs: `0.25%`, `4%`, `99%`
- Example of *invalid* inputs: `45 %`

13.2.27 SMB command

The SMB command used in the transaction. It can be a command id in decimal or hexadecimal form, or a command code inside strings.

Operators: !=, =

- Example of *valid* inputs: `'SMB2_com_logoff'`, `2`, `0x0e`
- Example of *invalid* inputs: `random text`

13.2.28 SMB status

The status of the SMB transaction. It can be a status id in decimal or hexadecimal form, or a status code inside quotes. The special values 'ok', 'warning' and 'error' are also accepted and mean, respectively, a match on every success, warning and error status. The special value 'common' matches a set of common statuses.

Operators: !=, =

- Example of *valid* inputs: `'SMB_status_no_such_file'`, `'error'`, `0xc000000f`
- Example of *invalid* inputs: `random text`

13.2.29 SMB sub-command

The SMB sub-command associated with the command used in the transaction. It can be a sub-command id in decimal or hexadecimal form, or a sub-command code inside strings.

Operators: !=, =

- Example of *valid* inputs: `'SMB_TRANS2_open2'`, `16`, `0x0d`
- Example of *invalid* inputs: `random text`

13.2.30 SQL command

A single SQL command, inside quotes.

Operators: !=, =

- Example of *valid* inputs: `'CREATE INDEX'`, `'INSERT'`
- Example of *invalid* inputs: `'SELECT * FROM users;'`, `INSERT`

13.2.31 SQL system

The name of the RDBMS dialect used in the connection, inside quotes.

Operators: `!=`, `=`

- Example of *valid* inputs: `'MySQL','TNS'`
- Example of *invalid* inputs: `MySQL`

13.2.32 String

A character string enclosed between single or double quotes. It can contains wildcards `*` that matches anything, or for more accurate search, it can be prefixed by a `~` which will treat the value as a regular expression pattern.

Operators: `!=`, `=`

- Example of *valid* inputs: `"*some thing*",'~^.*\.[a-z]{2}$'`
- Example of *invalid* inputs: `not enclosed between quotes`

13.2.33 Wildcard or regex

Either a string containing wildcards `*`, or a regular expression if prefixed by `~`. The value should be surrounded by simple or double quotes.

Operators: `!=`, `=`

- Example of *valid* inputs: `"google.com"`, `"~.*\.google\.(com|fr)"`, `'*.securactive.org'`
- Example of *invalid* inputs: `foo.com`

13.2.34 Zone name

The name of a zone, using the path notation `'/Private/Local'`. The = operator will return results matching **only** this specific zone, whereas the `in` operator will also return results contained in children zones. Note that the value mustbe enclosed between single or double quotes.

Operators: `!=`, `=`, `in`

- Example of *valid* inputs: `/Private/Local`
- Example of *invalid* inputs: `/NonExistent/Zone`

13.3 SPV For Developpers

For developpers, it is possible to programmatically generate and retrieve the result pages as HTML or PDF.

13.3.1 Getting Data

To request a page, you will use the same URL that you would in a Web browser. Filters are implemented in the URL in the form of GET parameters: `http://DOMAIN/PATH?filter.field1=value1&filter.field2=value2`

If the `capture_begin` or `capture_end` filters are omitted, the engine will instead request data for the last hour.

The URL can be parameterized to ask the engine to render the output either as stripped-down HTML, or as PDF. This is done by prepending, respectively,

`/++skin++simplehtml/` and `/++skin++pdf/` to the path part of the URL: `http://DOMAIN/++skin++pdf/PATH?filter.field1=value1&filter.field2=value2`

NB: This is the same kind of URL you get when you click the 'Export as PDF' button in the user interface.

13.3.2 Authentication

SPV normally uses its own authentication forms, which you see whenever you log in the user interface with a Web browser. This authentication system uses cookie-based sessions to keep you logged in, which can be inconvenient to support programmatically.

SPV therefore also provides support for session-less access with the Basic HTTP Authentication mechanism. Command-line download clients like `curl` or `wget` support it natively.

There are two ways to switch SPV over to session-less Basic HTTP Authentication:

1. Your download client may support it automatically. `curl` does. `wget` does if you pass the `--auth-no-challenge` command-line option. In this case, pass your login and password in the normal way supported by your client, and SPV will automatically authenticate your request using Basic HTTP Authentication.
2. Your download client may require the server itself to initiate the Basic HTTP Authentication process. For instance, `wget` does so when you omit the `--auth-no-challenge` option. If so, you can instruct SPV to initiate the process by appending `&auth=force-http` to the query string part of the URLs.

NB: Basic HTTP Authentication does not protect your credentials from snooping. You may thus want to use `https://` URLs instead of `http://`.

13.3.3 Scripting Examples

In the first example, we will retrieve the *Top Servers* page as stripped-down HTML, filtering for the *SSH* application, using the command-line with `wget`:

```
## Using the --auth-no-challenge option:
wget --user=admin --password=admin --auth-no-challenge 'http://SPV/++skin++simplehtml/nevrax/netwo

## Using the *auth* query string parameter:
wget --user=admin --password=admin 'http://SPV/++skin++simplehtml/nevrax/network/ipstats_dst.html
```

In the next example, we will retrieve the *Bandwidth Chart* page as a PDF, using the command-line with `curl`:

```
## curl will automatically initiate Basic HTTP Authentication when you pass credentials with the
curl -u admin:admin 'http://SPV/++skin++pdf/nevrax/network/bw_chart_page.html?filter.capture_begi
```

If HTTPS is used to keep your credentials concealed, your client may need an option to skip the server certificate check. Here is an example with `wget`:

```
## Same wget query as above, using HTTPS, and a terser way to pass credentials:
wget --no-check-certificate 'https://admin:admin/SPV/++skin++simplehtml/nevrax/network/ipstats_dst
```

13.3.4 Programming Example

You can also create a program to retrieve result pages from SPV. Here is a simple example in *python2*:

```
import urllib
import urllib2

def get_spv_data(url, user, passw):

    ## create authentication
    auth = urllib2.HTTPPasswordMgrWithDefaultRealm()
```

```
    auth.add_password(None, url, user, passw)
    urllib2.install_opener(urllib2.build_opener(urllib2.HTTPBasicAuthHandler(auth)))

    ## request
    req = urllib2.Request(url)
    f = urllib2.urlopen(req)
    return f.read()


def create_url(domain, page, filters={}, as_pdf=True):
    filter_args = {('filter.%s' % k): v for k, v in filters.iteritems()}
    filter_args = urllib.urlencode(filter_args)
    if as_pdf:
        skin = '++skin++pdf/'
    else:
        skin = '++skin++simplehtml/'
    return 'http://%s/%s%s?%s&auth=force-http' % (domain, skin, page, filter_args)


## set up the query
domain = 'myspv-domain'
page = 'nevrax/network/bw_chart_page.html'
filters = {'capture_begin': '2013-01-31 14:50',
           'capture_end':   '2013-01-31 15:50',
           'serviceid':     'ssh',
           }
user = 'admin'
passw = 'admin'

url = create_url(domain, page, filters, as_pdf=True)
result = get_spv_data(url, user, passw)
open("output.pdf", "w").write(result)
```

13.4 Protocol Stack List

The SPV sniffer decodes the different protocol levels, as described in the Glossary (*Protocol Stack*).

You can use them to filter the captured flows or to define applications.

Here the comprehensive list of them, with an example of a common use case:

Protostack	Description	Common Use Case
ARP	Address Resolution Protocol	Ethernet/ARP
BGP	Border Gateway Protocol	Ethernet/BGP
Bittorrent	Peer to peer file sharing	Ethernet/IPv4/TCP/Bittorent
CIFS	Microsoft Common Internet File System	Ethernet/IPv4/TCP/Netbios/CIFS
Citrix	[BETA] Citrix Remote Desktop	Ethernet/IPv4/TCP/Citrix
DHCP	Dynamic Host Configuration Protocol	Ethernet/IPv4/UDP/DHCP
DNS	Domain Name System	Ethernet/IPv4/UDP/DNS
ERSPAN	Cisco Encapsulated Remote Switched Port Analyzer	Ethernet/Ipv4/GRE/ERSPAN/...
Ethernet	Ethernet Protocol. Certainly the first protocol of the stack	Ethernet/...
FCoE	Fibre Channel over Ethernet	Ethernet/FCoE/...
FTP	File Transfer Protocol	Ethernet/IPv4/TCP/FTP
Gnutella	Peer to peer file sharing	Ethernet/IPv4/UDP/Gnutella
GRE	Generic Routing Encapsulation (tunneling)	Ethernet/IPv4/GRE/...
HTTP	Hypertext Transfer Protocol	Ethernet/IPv4/TCP/TLS/HTTP
ICMP	Internet Control Message Protocol	Ethernet/IPv4/ICMP
ICMPv6	Internet Control Message Protocol for IPv6	Ethernet/IPv6/ICMPv6

Continued on next page

Table 13.9 – continued from previous page

Protostack	Description	Common Use Case
IMAP	Internet Message Access Protocol	Ethernet/IPv4/TCP/IMAP
IPv4	Internet Protocol version 4	Ethernet/IPv4/...
IPv6	Internet Protocol version 6	Ethernet/IPv6/...
IRC	Internet Relay Chat	Ethernet/IPv4/TCP/IRC
Jabber	Extensible Messaging and Presence Protocol	Ethernet/IPv4/TCP/Jabber
MGCP	Media Gateway Control Protocol	Ethernet/IPv4/UDP/MGCP/SDP
MySQL	MySQL or MariaDB databases	Ethernet/IPv4/TCP/MySQL
Netbios	Network Basic Input/Output System	Ethernet/IPv4/TCP/Netbios/...
NTP	Network Time Protocol	Ethernet/IPv6/UDP/NTP
PCanywhere	Symantec's PCanywhere	Ethernet/IPv4/TCP/PCanywhere
POP	Post Office Protocol	Ethernet/IPv6/TCP/POP
PostgreSQL	PostgreSQL database	Ethernet/IPv4/TCP/PostgreSQL
RDP	Remote Desktop Protocol	Ethernet/IPv4/TCP/RDP
RTCP	RTP Control Protocol	Ethernet/IPv4/UDP/RTCP
RTP	Real-time Transport Protocol	Ethernet/IPv4/UDP/RTP
SDP	Session Description Protocol	Ethernet/IPv4/UDP/SIP/SDP
SIP	Session Initiation Protocol	Ethernet/IPv4/UDP/SIP
SKINNY	Skinny Client Control Protocol	Ethernet/IPv4/TCP/SKINNY
SMTP	Simple Mail Transfer Protocol	Ethernet/IPv4/TCP/SMTP
SSLv2	Secure Sockets Layer	Ethernet/IPv4/TCP/SSLv2
TCP	Transmission Control Protocol	Ethernet/IPv4/TCP/...
TDS	Tabular Data Stream	Ethernet/IPv4/TCP/TDS/...
TDS(msg)	Tabular Data Stream messages	Ethernet/IPv4/TCP/TDS/TDS(msg)
Telnet	Interactive terminal	Ethernet/IPv4/TCP/Telnet
TLS	Transport Layer Security	Ethernet/IPv4/TCP/TLS/...
TNS	Transparent Network Substrate (Oracle)	Ethernet/IPv4/TCP/TNS
UDP	User Datagram Protocol	Ethernet/IPv4/UDP/...
VNC	Virtual Network Computing	Ethernet/IPv4/TCP/VNC

13.5 Licenses of open-source libraries

13.5.1 Operating System:

SPV uses the Debian operating system (http://www.debian.org) SPV does not use the 'non-free' repository provided by Debian.

According to Debian Social Contract v1.1 (April 26, 2004): the license of a Debian component may not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license may not require a royalty or other fee for such sale.

13.5.2 License inventory for the sniffer:

- glibc : GNU Lesser General Public License.
- **guile library** [GNU General Public License, version 2 or later] with exceptions (no copyleft on link).
- guile-sqlite3 : GNU Lesser General Public License, version 3 or later.
- Junkie : Copyright and AGPLv3
- libpcap library : BSD style license (3-clause BSD)
- libgc : libgc license
- libuuid : libuuid License

- openssl library : BSD style license + SSLeay license
- p0f.fp file : GNU Lesser General Public License.

13.5.3 License inventory for the javascript GUI:

- jquery : The MIT License
- jquery.mb.browser : The MIT License or GNU General Public License
- jquery.multiple.select : The MIT License
- jquery.ui : The MIT License
- jquery.ui.statusbar : The MIT License
- mColorPicker : The MIT License

13.5.4 License inventory for GUI and database management:

- jsonrpclib : Apache
- Salt : Apache
- lxml : BSD / GPL / PSF / CWI
- enum : Choice of GPL or Python license
- psycopg2 : GPL with exceptions or ZPL
- dnspython : ISC
- paramiko : LGPL
- PostgreSQL : PostgreSQL Licence
- setuptools : PSF
- FormEncode : PSF
- pycrypto : Public domain
- docutils : public domain, Python, 2-Clause BSD, GPL 3 (see COPYING.txt)
- Python : Python Software Foundation licence
- pychartdir : SecurActive license
- Pillow : Standard PIL License

Components licensed under the 'Repoze' license:

- Chameleon
- pyramid
- pyramid_mako
- pyramid_tm
- repoze.lru
- repoze.profile
- repoze.who
- superlance
- supervisor

- translationstring
- venusian

Components licensed under the 'BSD' license:

- Fanstatic
- Jinja2
- MarkupSafe
- Pygments
- Sphinx
- collective.recipe.template
- configobj
- fanstatic
- js.amcharts
- js.jquery
- js.jquery_kinetic
- js.jquery_timepicker_addon
- js.jqueryui
- mechanize
- mock
- netaddr
- pyprof2calltree
- pyramid_debugtoolbar
- reportlab
- sqlparse

Components licensed under the 'MIT' license:

- Mailer
- Mako
- MiniMock
- Paste
- PasteDeploy
- SQLAlchemy
- Tempita
- WebOb
- WebTest
- beautifulsoup4
- cmdln
- ecdsa
- pyparsing

- pytz
- six
- wsgi_intercept

13.6 CIFS Status Categories

The CIFS Status codes must be interpreted in one of two ways, depending on the capabilities negotiated between the client and the server: either as an NTSTATUS value, or as an SMBSTATUS value. For more details on the NT statuses, you can check the official documentation here (http://msdn.microsoft.com/en-us/library/cc704588.aspx). If you're looking for the SMB statuses, check the CIFS documentation here (http://msdn.microsoft.com/en-us/library/ee441884.aspx).

We have classified these statuses into three categories, depending on their severity, as shown in the table below.

Code	Status	NTSTATUS severity	S
0x00000000	SMB_STATUS_OK		C
0x00000080	NT_STATUS_ABANDONED	OK	
0x000000c0	NT_STATUS_USER_APC	OK	
0x00000100	NT_STATUS_KERNEL_APC	OK	
0x00000101	NT_STATUS_ALERTED	OK	
0x00000102	NT_STATUS_TIMEOUT	OK	
0x00000103	NT_STATUS_PENDING	OK	
0x00000104	NT_STATUS_REPARSE	OK	
0x00000105	NT_STATUS_MORE_ENTRIES	OK	
0x00000106	NT_STATUS_NOT_ALL_ASSIGNED	OK	
0x00000107	NT_STATUS_SOME_NOT_MAPPED	OK	
0x00000108	NT_STATUS_OPLOCK_BREAK_IN_PROGRESS	OK	
0x00000109	NT_STATUS_VOLUME_MOUNTED	OK	
0x0000010a	NT_STATUS_RXACT_COMMITTED	OK	
0x0000010b	NT_STATUS_NOTIFY_CLEANUP	OK	
0x0000010c	SMB_STATUS_NOTIFY_ENUM_DIR		W
0x0000010d	NT_STATUS_NO_QUOTAS_FOR_ACCOUNT	OK	
0x0000010e	NT_STATUS_PRIMARY_TRANSPORT_CONNECT_FAILED	OK	
0x00000110	NT_STATUS_PAGE_FAULT_TRANSITION	OK	
0x00000111	NT_STATUS_PAGE_FAULT_DEMAND_ZERO	OK	
0x00000112	NT_STATUS_PAGE_FAULT_COPY_ON_WRITE	OK	
0x00000113	NT_STATUS_PAGE_FAULT_GUARD_PAGE	OK	
0x00000114	NT_STATUS_PAGE_FAULT_PAGING_FILE	OK	
0x00000115	NT_STATUS_CACHE_PAGE_LOCKED	OK	
0x00000116	NT_STATUS_CRASH_DUMP	OK	
0x00000117	NT_STATUS_BUFFER_ALL_ZEROS	OK	
0x00000118	NT_STATUS_REPARSE_OBJECT	OK	
0x00000119	NT_STATUS_RESOURCE_REQUIREMENTS_CHANGED	OK	
0x00000120	NT_STATUS_TRANSLATION_COMPLETE	OK	
0x00000121	NT_STATUS_DS_MEMBERSHIP_EVALUATED_LOCALLY	OK	
0x00000122	NT_STATUS_NOTHING_TO_TERMINATE	OK	
0x00000123	NT_STATUS_PROCESS_NOT_IN_JOB	OK	
0x00000124	NT_STATUS_PROCESS_IN_JOB	OK	
0x00000367	NT_STATUS_WAIT_FOR_OPLOCK	OK	
0x00010002	SMB_STATUS_INVALID_SMB		W
0x00050002	SMB_STATUS_SMB_BAD_TID		W
0x00060001	SMB_STATUS_SMB_BAD_FID		W
0x000c0001	SMB_STATUS_OS2_INVALID_ACCESS		W
0x00160002	SMB_STATUS_SMB_BAD_COMMAND		W
0x005b0002	SMB_STATUS_SMB_BAD_UID		W

Co

Table 13.10 – continued from previous page

Code	Status	NTSTATUS severity	S
0x00710001	SMB_STATUS_OS2_NO_MORE_SIDS		W
0x007c0001	SMB_STATUS_OS2_INVALID_LEVEL		W
0x00830001	SMB_STATUS_OS2_NEGATIVE_SEEK		W
0x00ad0001	SMB_STATUS_OS2_CANCEL_VIOLATION		W
0x00ae0001	SMB_STATUS_OS2_ATOMIC_LOCKS_NOT_SUPPORTED		W
0x00fa0002	SMB_STATUS_SMB_USE_MPX		W
0x00fb0002	SMB_STATUS_SMB_USE_STANDARD		W
0x00fc0002	SMB_STATUS_SMB_CONTINUE_MPX		W
0x010a0001	SMB_STATUS_OS2_CANNOT_COPY		W
0x01130001	SMB_STATUS_OS2_EAS_DIDNT_FIT		W
0x03e20001	SMB_STATUS_OS2_EA_ACCESS_DENIED		W
0x40000000	NT_STATUS_OBJECT_NAME_EXISTS	OK	
0x40000001	NT_STATUS_THREAD_WAS_SUSPENDED	OK	
0x40000002	NT_STATUS_WORKING_SET_LIMIT_RANGE	OK	
0x40000003	NT_STATUS_IMAGE_NOT_AT_BASE	OK	
0x40000004	NT_STATUS_RXACT_STATE_CREATED	OK	
0x40000005	NT_STATUS_SEGMENT_NOTIFICATION	OK	
0x40000006	NT_STATUS_LOCAL_USER_SESSION_KEY	OK	
0x40000007	NT_STATUS_BAD_CURRENT_DIRECTORY	OK	
0x40000008	NT_STATUS_SERIAL_MORE_WRITES	OK	
0x40000009	NT_STATUS_REGISTRY_RECOVERED	OK	
0x4000000a	NT_STATUS_FT_READ_RECOVERY_FROM_BACKUP	OK	
0x4000000b	NT_STATUS_FT_WRITE_RECOVERY	OK	
0x4000000c	NT_STATUS_SERIAL_COUNTER_TIMEOUT	OK	
0x4000000d	NT_STATUS_NULL_LM_PASSWORD	OK	
0x4000000e	NT_STATUS_IMAGE_MACHINE_TYPE_MISMATCH	OK	
0x4000000f	NT_STATUS_RECEIVE_PARTIAL	OK	
0x40000010	NT_STATUS_RECEIVE_EXPEDITED	OK	
0x40000011	NT_STATUS_RECEIVE_PARTIAL_EXPEDITED	OK	
0x40000012	NT_STATUS_EVENT_DONE	OK	
0x40000013	NT_STATUS_EVENT_PENDING	OK	
0x40000014	NT_STATUS_CHECKING_FILE_SYSTEM	OK	
0x40000015	NT_STATUS_FATAL_APP_EXIT	OK	
0x40000016	NT_STATUS_PREDEFINED_HANDLE	OK	
0x40000017	NT_STATUS_WAS_UNLOCKED	OK	
0x40000018	NT_STATUS_SERVICE_NOTIFICATION	OK	
0x40000019	NT_STATUS_WAS_LOCKED	OK	
0x4000001a	NT_STATUS_LOG_HARD_ERROR	OK	
0x4000001b	NT_STATUS_ALREADY_WIN32	OK	
0x4000001c	NT_STATUS_WX86_UNSIMULATE	OK	
0x4000001d	NT_STATUS_WX86_CONTINUE	OK	
0x4000001e	NT_STATUS_WX86_SINGLE_STEP	OK	
0x4000001f	NT_STATUS_WX86_BREAKPOINT	OK	
0x40000020	NT_STATUS_WX86_EXCEPTION_CONTINUE	OK	
0x40000021	NT_STATUS_WX86_EXCEPTION_LASTCHANCE	OK	
0x40000022	NT_STATUS_WX86_EXCEPTION_CHAIN	OK	
0x40000023	NT_STATUS_IMAGE_MACHINE_TYPE_MISMATCH_EXE	OK	
0x40000024	NT_STATUS_NO_YIELD_PERFORMED	OK	
0x40000025	NT_STATUS_TIMER_RESUME_IGNORED	OK	
0x40000026	NT_STATUS_ARBITRATION_UNHANDLED	OK	
0x40000027	NT_STATUS_CARDBUS_NOT_SUPPORTED	OK	
0x40000028	NT_STATUS_WX86_CREATEWX86TIB	OK	
0x40000029	NT_STATUS_MP_PROCESSOR_MISMATCH	OK	
0x4000002a	NT_STATUS_HIBERNATED	OK	
0x4000002b	NT_STATUS_RESUME_HIBERNATION	OK	

Co

Table 13.10 – continued from previous page

Code	Status	NTSTATUS severity	S
0x40000294	NT_STATUS_WAKE_SYSTEM	OK	
0x40000370	NT_STATUS_DS_SHUTTING_DOWN	OK	
0x40020056	RPC_NT_UUID_LOCAL_ONLY	OK	
0x400200af	RPC_NT_SEND_INCOMPLETE	OK	
0x80000001	NT_STATUS_GUARD_PAGE_VIOLATION	WARNING	
0x80000002	NT_STATUS_DATATYPE_MISALIGNMENT	WARNING	
0x80000003	NT_STATUS_BREAKPOINT	WARNING	
0x80000004	NT_STATUS_SINGLE_STEP	WARNING	
0x80000005	SMB_STATUS_BUFFER_OVERFLOW		W
0x80000006	SMB_STATUS_NO_MORE_FILES		W
0x80000007	NT_STATUS_WAKE_SYSTEM_DEBUGGER	WARNING	
0x8000000a	NT_STATUS_HANDLES_CLOSED	WARNING	
0x8000000b	NT_STATUS_NO_INHERITANCE	WARNING	
0x8000000c	NT_STATUS_GUID_SUBSTITUTION_MADE	WARNING	
0x8000000d	NT_STATUS_PARTIAL_COPY	WARNING	
0x8000000e	SMB_STATUS_DEVICE_PAPER_EMPTY		E
0x8000000f	NT_STATUS_DEVICE_POWERED_OFF	WARNING	
0x80000010	NT_STATUS_DEVICE_OFF_LINE	WARNING	
0x80000011	NT_STATUS_DEVICE_BUSY	WARNING	
0x80000012	NT_STATUS_NO_MORE_EAS	WARNING	
0x80000013	NT_STATUS_INVALID_EA_NAME	WARNING	
0x80000014	NT_STATUS_EA_LIST_INCONSISTENT	WARNING	
0x80000015	NT_STATUS_INVALID_EA_FLAG	WARNING	
0x80000016	NT_STATUS_VERIFY_REQUIRED	WARNING	
0x80000017	NT_STATUS_EXTRANEOUS_INFORMATION	WARNING	
0x80000018	NT_STATUS_RXACT_COMMIT_NECESSARY	WARNING	
0x8000001a	NT_STATUS_NO_MORE_ENTRIES	WARNING	
0x8000001b	NT_STATUS_FILEMARK_DETECTED	WARNING	
0x8000001c	NT_STATUS_MEDIA_CHANGED	WARNING	
0x8000001d	NT_STATUS_BUS_RESET	WARNING	
0x8000001e	NT_STATUS_END_OF_MEDIA	WARNING	
0x8000001f	NT_STATUS_BEGINNING_OF_MEDIA	WARNING	
0x80000020	NT_STATUS_MEDIA_CHECK	WARNING	
0x80000021	NT_STATUS_SETMARK_DETECTED	WARNING	
0x80000022	NT_STATUS_NO_DATA_DETECTED	WARNING	
0x80000023	NT_STATUS_REDIRECTOR_HAS_OPEN_HANDLES	WARNING	
0x80000024	NT_STATUS_SERVER_HAS_OPEN_HANDLES	WARNING	
0x80000025	NT_STATUS_ALREADY_DISCONNECTED	WARNING	
0x80000026	NT_STATUS_LONGJUMP	WARNING	
0x80000027	NT_STATUS_CLEANER_CARTRIDGE_INSTALLED	WARNING	
0x80000028	NT_STATUS_PLUGPLAY_QUERY_VETOED	WARNING	
0x80000029	NT_STATUS_UNWIND_CONSOLIDATE	WARNING	
0x8000002d	STATUS_STOPPED_ON_SYMLINK	WARNING	
0x80000288	NT_STATUS_DEVICE_REQUIRES_CLEANING	WARNING	
0x80000289	NT_STATUS_DEVICE_DOOR_OPEN	WARNING	
0xc0000001	SMB_STATUS_UNSUCCESSFUL		W
0xc0000002	SMB_STATUS_NOT_IMPLEMENTED		W
0xc0000003	SMB_STATUS_INVALID_INFO_CLASS		W
0xc0000004	NT_STATUS_INFO_LENGTH_MISMATCH	ERROR	
0xc0000005	NT_STATUS_ACCESS_VIOLATION	ERROR	
0xc0000006	NT_STATUS_IN_PAGE_ERROR	ERROR	
0xc0000007	NT_STATUS_PAGEFILE_QUOTA	ERROR	
0xc0000008	SMB_STATUS_INVALID_HANDLE		W
0xc0000009	NT_STATUS_BAD_INITIAL_STACK	ERROR	
0xc000000a	NT_STATUS_BAD_INITIAL_PC	ERROR	

Co

Table 13.10 – continued from previous page

Code	Status	NTSTATUS severity	S
0xc000000b	NT_STATUS_INVALID_CID	ERROR	
0xc000000c	NT_STATUS_TIMER_NOT_CANCELED	ERROR	
0xc000000d	SMB_STATUS_INVALID_PARAMETER		V
0xc000000e	SMB_STATUS_NO_SUCH_DEVICE		V
0xc000000f	SMB_STATUS_NO_SUCH_FILE		V
0xc0000010	SMB_STATUS_INVALID_DEVICE_REQUEST		V
0xc0000011	SMB_STATUS_END_OF_FILE		V
0xc0000012	SMB_STATUS_WRONG_VOLUME		E
0xc0000013	SMB_STATUS_NO_MEDIA_IN_DEVICE		E
0xc0000014	NT_STATUS_UNRECOGNIZED_MEDIA	ERROR	
0xc0000015	SMB_STATUS_NONEXISTENT_SECTOR		E
0xc0000016	SMB_STATUS_MORE_PROCESSING_REQUIRED		V
0xc0000017	NT_STATUS_NO_MEMORY	ERROR	
0xc0000018	NT_STATUS_CONFLICTING_ADDRESSES	ERROR	
0xc0000019	NT_STATUS_NOT_MAPPED_VIEW	ERROR	
0xc000001a	NT_STATUS_UNABLE_TO_FREE_VM	ERROR	
0xc000001b	NT_STATUS_UNABLE_TO_DELETE_SECTION	ERROR	
0xc000001c	NT_STATUS_INVALID_SYSTEM_SERVICE	ERROR	
0xc000001d	NT_STATUS_ILLEGAL_INSTRUCTION	ERROR	
0xc000001e	SMB_STATUS_INVALID_LOCK_SEQUENCE		V
0xc000001f	SMB_STATUS_INVALID_VIEW_SIZE		V
0xc0000020	NT_STATUS_INVALID_FILE_FOR_SECTION	ERROR	
0xc0000021	SMB_STATUS_ALREADY_COMMITTED		V
0xc0000022	SMB_STATUS_ACCESS_DENIED		V
0xc0000023	NT_STATUS_BUFFER_TOO_SMALL	ERROR	
0xc0000024	SMB_STATUS_OBJECT_TYPE_MISMATCH		V
0xc0000025	NT_STATUS_NONCONTINUABLE_EXCEPTION	ERROR	
0xc0000026	NT_STATUS_INVALID_DISPOSITION	ERROR	
0xc0000027	NT_STATUS_UNWIND	ERROR	
0xc0000028	NT_STATUS_BAD_STACK	ERROR	
0xc0000029	NT_STATUS_INVALID_UNWIND_TARGET	ERROR	
0xc000002a	NT_STATUS_NOT_LOCKED	ERROR	
0xc000002b	NT_STATUS_PARITY_ERROR	ERROR	
0xc000002c	NT_STATUS_UNABLE_TO_DECOMMIT_VM	ERROR	
0xc000002d	NT_STATUS_NOT_COMMITTED	ERROR	
0xc000002e	NT_STATUS_INVALID_PORT_ATTRIBUTES	ERROR	
0xc000002f	NT_STATUS_PORT_MESSAGE_TOO_LONG	ERROR	
0xc0000030	NT_STATUS_INVALID_PARAMETER_MIX	ERROR	
0xc0000031	NT_STATUS_INVALID_QUOTA_LOWER	ERROR	
0xc0000032	SMB_STATUS_DISK_CORRUPT_ERROR		E
0xc0000033	NT_STATUS_OBJECT_NAME_INVALID	ERROR	
0xc0000034	SMB_STATUS_OBJECT_NAME_NOT_FOUND		V
0xc0000035	SMB_STATUS_OBJECT_NAME_COLLISION		V
0xc0000037	SMB_STATUS_PORT_DISCONNECTED		V
0xc0000038	NT_STATUS_DEVICE_ALREADY_ATTACHED	ERROR	
0xc0000039	SMB_STATUS_OBJECT_PATH_INVALID		V
0xc000003a	SMB_STATUS_OBJECT_PATH_NOT_FOUND		V
0xc000003b	SMB_STATUS_OBJECT_PATH_SYNTAX_BAD		V
0xc000003c	NT_STATUS_DATA_OVERRUN	ERROR	
0xc000003d	NT_STATUS_DATA_LATE_ERROR	ERROR	
0xc000003e	SMB_STATUS_DATA_ERROR		E
0xc000003f	SMB_STATUS_CRC_ERROR		E
0xc0000040	SMB_STATUS_SECTION_TOO_BIG		V
0xc0000041	SMB_STATUS_PORT_CONNECTION_REFUSED		V
0xc0000042	SMB_STATUS_INVALID_PORT_HANDLE		V

Co

Table 13.10 – continued from previous page

Code	Status	NTSTATUS severity	S
0xc0000043	SMB_STATUS_SHARING_VIOLATION		E
0xc0000044	NT_STATUS_QUOTA_EXCEEDED	ERROR	
0xc0000045	NT_STATUS_INVALID_PAGE_PROTECTION	ERROR	
0xc0000046	NT_STATUS_MUTANT_NOT_OWNED	ERROR	
0xc0000047	NT_STATUS_SEMAPHORE_LIMIT_EXCEEDED	ERROR	
0xc0000048	NT_STATUS_PORT_ALREADY_SET	ERROR	
0xc0000049	NT_STATUS_SECTION_NOT_IMAGE	ERROR	
0xc000004a	NT_STATUS_SUSPEND_COUNT_EXCEEDED	ERROR	
0xc000004b	SMB_STATUS_THREAD_IS_TERMINATING		W
0xc000004c	NT_STATUS_BAD_WORKING_SET_LIMIT	ERROR	
0xc000004d	NT_STATUS_INCOMPATIBLE_FILE_MAP	ERROR	
0xc000004e	NT_STATUS_SECTION_PROTECTION	ERROR	
0xc000004f	SMB_STATUS_EAS_NOT_SUPPORTED		W
0xc0000050	SMB_STATUS_EA_TOO_LARGE		W
0xc0000051	NT_STATUS_NONEXISTENT_EA_ENTRY	ERROR	
0xc0000052	NT_STATUS_NO_EAS_ON_FILE	ERROR	
0xc0000053	NT_STATUS_EA_CORRUPT_ERROR	ERROR	
0xc0000054	SMB_STATUS_FILE_LOCK_CONFLICT		E
0xc0000055	SMB_STATUS_LOCK_NOT_GRANTED		W
0xc0000056	SMB_STATUS_DELETE_PENDING		W
0xc0000057	NT_STATUS_CTL_FILE_NOT_SUPPORTED	ERROR	
0xc0000058	NT_STATUS_UNKNOWN_REVISION	ERROR	
0xc0000059	NT_STATUS_REVISION_MISMATCH	ERROR	
0xc000005a	NT_STATUS_INVALID_OWNER	ERROR	
0xc000005b	NT_STATUS_INVALID_PRIMARY_GROUP	ERROR	
0xc000005c	NT_STATUS_NO_IMPERSONATION_TOKEN	ERROR	
0xc000005d	NT_STATUS_CANT_DISABLE_MANDATORY	ERROR	
0xc000005e	NT_STATUS_NO_LOGON_SERVERS	ERROR	
0xc000005f	NT_STATUS_NO_SUCH_LOGON_SESSION	ERROR	
0xc0000060	NT_STATUS_NO_SUCH_PRIVILEGE	ERROR	
0xc0000061	SMB_STATUS_PRIVILEGE_NOT_HELD		W
0xc0000062	NT_STATUS_INVALID_ACCOUNT_NAME	ERROR	
0xc0000063	NT_STATUS_USER_EXISTS	ERROR	
0xc0000064	NT_STATUS_NO_SUCH_USER	ERROR	
0xc0000065	NT_STATUS_GROUP_EXISTS	ERROR	
0xc0000066	NT_STATUS_NO_SUCH_GROUP	ERROR	
0xc0000067	NT_STATUS_MEMBER_IN_GROUP	ERROR	
0xc0000068	NT_STATUS_MEMBER_NOT_IN_GROUP	ERROR	
0xc0000069	NT_STATUS_LAST_ADMIN	ERROR	
0xc000006a	SMB_STATUS_WRONG_PASSWORD		W
0xc000006b	NT_STATUS_ILL_FORMED_PASSWORD	ERROR	
0xc000006c	NT_STATUS_PASSWORD_RESTRICTION	ERROR	
0xc000006d	SMB_STATUS_LOGON_FAILURE		W
0xc000006e	NT_STATUS_ACCOUNT_RESTRICTION	ERROR	
0xc000006f	SMB_STATUS_INVALID_LOGON_HOURS		W
0xc0000070	SMB_STATUS_INVALID_WORKSTATION		W
0xc0000071	SMB_STATUS_PASSWORD_EXPIRED		W
0xc0000072	SMB_STATUS_ACCOUNT_DISABLED		W
0xc0000074	NT_STATUS_TOO_MANY_LUIDS_REQUESTED	ERROR	
0xc0000075	NT_STATUS_LUIDS_EXHAUSTED	ERROR	
0xc0000076	NT_STATUS_INVALID_SUB_AUTHORITY	ERROR	
0xc0000077	NT_STATUS_INVALID_ACL	ERROR	
0xc0000078	NT_STATUS_INVALID_SID	ERROR	
0xc0000079	NT_STATUS_INVALID_SECURITY_DESCR	ERROR	
0xc000007a	NT_STATUS_PROCEDURE_NOT_FOUND	ERROR	

Co

Table 13.10 – continued from previous page

Code	Status	NTSTATUS severity	S
0xc000007b	NT_STATUS_INVALID_IMAGE_FORMAT	ERROR	
0xc000007c	NT_STATUS_NO_TOKEN	ERROR	
0xc000007d	NT_STATUS_BAD_INHERITANCE_ACL	ERROR	
0xc000007e	SMB_STATUS_RANGE_NOT_LOCKED		W
0xc000007f	SMB_STATUS_DISK_FULL		E
0xc0000080	NT_STATUS_SERVER_DISABLED	ERROR	
0xc0000081	NT_STATUS_SERVER_NOT_DISABLED	ERROR	
0xc0000082	NT_STATUS_TOO_MANY_GUIDS_REQUESTED	ERROR	
0xc0000083	NT_STATUS_GUIDS_EXHAUSTED	ERROR	
0xc0000084	NT_STATUS_INVALID_ID_AUTHORITY	ERROR	
0xc0000085	NT_STATUS_AGENTS_EXHAUSTED	ERROR	
0xc0000086	NT_STATUS_INVALID_VOLUME_LABEL	ERROR	
0xc0000087	NT_STATUS_SECTION_NOT_EXTENDED	ERROR	
0xc0000088	NT_STATUS_NOT_MAPPED_DATA	ERROR	
0xc0000089	NT_STATUS_RESOURCE_DATA_NOT_FOUND	ERROR	
0xc000008a	NT_STATUS_RESOURCE_TYPE_NOT_FOUND	ERROR	
0xc000008b	NT_STATUS_RESOURCE_NAME_NOT_FOUND	ERROR	
0xc000008c	NT_STATUS_ARRAY_BOUNDS_EXCEEDED	ERROR	
0xc000008d	NT_STATUS_FLOAT_DENORMAL_OPERAND	ERROR	
0xc000008e	NT_STATUS_FLOAT_DIVIDE_BY_ZERO	ERROR	
0xc000008f	NT_STATUS_FLOAT_INEXACT_RESULT	ERROR	
0xc0000090	NT_STATUS_FLOAT_INVALID_OPERATION	ERROR	
0xc0000091	NT_STATUS_FLOAT_OVERFLOW	ERROR	
0xc0000092	NT_STATUS_FLOAT_STACK_CHECK	ERROR	
0xc0000093	NT_STATUS_FLOAT_UNDERFLOW	ERROR	
0xc0000094	NT_STATUS_INTEGER_DIVIDE_BY_ZERO	ERROR	
0xc0000095	NT_STATUS_INTEGER_OVERFLOW	ERROR	
0xc0000096	NT_STATUS_PRIVILEGED_INSTRUCTION	ERROR	
0xc0000097	SMB_STATUS_TOO_MANY_PAGING_FILES		W
0xc0000098	NT_STATUS_FILE_INVALID	ERROR	
0xc0000099	NT_STATUS_ALLOTTED_SPACE_EXCEEDED	ERROR	
0xc000009a	NT_STATUS_INSUFFICIENT_RESOURCES	ERROR	
0xc000009b	SMB_STATUS_DFS_EXIT_PATH_FOUND		W
0xc000009c	SMB_STATUS_DATA_ERROR_UNUSED		W
0xc000009d	NT_STATUS_DEVICE_NOT_CONNECTED	ERROR	
0xc000009e	NT_STATUS_DEVICE_POWER_FAILURE	ERROR	
0xc000009f	NT_STATUS_FREE_VM_NOT_AT_BASE	ERROR	
0xc00000a0	NT_STATUS_MEMORY_NOT_ALLOCATED	ERROR	
0xc00000a1	NT_STATUS_WORKING_SET_QUOTA	ERROR	
0xc00000a2	SMB_STATUS_MEDIA_WRITE_PROTECTED		E
0xc00000a3	NT_STATUS_DEVICE_NOT_READY	ERROR	
0xc00000a4	NT_STATUS_INVALID_GROUP_ATTRIBUTES	ERROR	
0xc00000a5	NT_STATUS_BAD_IMPERSONATION_LEVEL	ERROR	
0xc00000a6	NT_STATUS_CANT_OPEN_ANONYMOUS	ERROR	
0xc00000a7	NT_STATUS_BAD_VALIDATION_CLASS	ERROR	
0xc00000a8	NT_STATUS_BAD_TOKEN_TYPE	ERROR	
0xc00000a9	NT_STATUS_BAD_MASTER_BOOT_RECORD	ERROR	
0xc00000aa	NT_STATUS_INSTRUCTION_MISALIGNMENT	ERROR	
0xc00000ab	SMB_STATUS_INSTANCE_NOT_AVAILABLE		W
0xc00000ac	SMB_STATUS_PIPE_NOT_AVAILABLE		W
0xc00000ad	SMB_STATUS_INVALID_PIPE_STATE		W
0xc00000ae	SMB_STATUS_PIPE_BUSY		W
0xc00000af	SMB_STATUS_ILLEGAL_FUNCTION		W
0xc00000b0	SMB_STATUS_PIPE_DISCONNECTED		W
0xc00000b1	SMB_STATUS_PIPE_CLOSING		W

Co

Table 13.10 – continued from previous page

Code	Status	NTSTATUS severity	S
0xc00000b2	NT_STATUS_PIPE_CONNECTED	ERROR	
0xc00000b3	NT_STATUS_PIPE_LISTENING	ERROR	
0xc00000b4	SMB_STATUS_INVALID_READ_MODE		V
0xc00000b5	SMB_STATUS_IO_TIMEOUT		V
0xc00000b6	NT_STATUS_FILE_FORCED_CLOSED	ERROR	
0xc00000b7	NT_STATUS_PROFILING_NOT_STARTED	ERROR	
0xc00000b8	NT_STATUS_PROFILING_NOT_STOPPED	ERROR	
0xc00000b9	NT_STATUS_COULD_NOT_INTERPRET	ERROR	
0xc00000ba	SMB_STATUS_FILE_IS_A_DIRECTORY		V
0xc00000bb	STATUS_NOT_SUPPORTED		V
0xc00000bc	NT_STATUS_REMOTE_NOT_LISTENING	ERROR	
0xc00000bd	NT_STATUS_DUPLICATE_NAME	ERROR	
0xc00000be	NT_STATUS_BAD_NETWORK_PATH	ERROR	
0xc00000bf	NT_STATUS_NETWORK_BUSY	ERROR	
0xc00000c0	NT_STATUS_DEVICE_DOES_NOT_EXIST	ERROR	
0xc00000c1	NT_STATUS_TOO_MANY_COMMANDS	ERROR	
0xc00000c2	NT_STATUS_ADAPTER_HARDWARE_ERROR	ERROR	
0xc00000c3	NT_STATUS_INVALID_NETWORK_RESPONSE	ERROR	
0xc00000c4	SMB_STATUS_UNEXPECTED_NETWORK_ERROR		V
0xc00000c5	NT_STATUS_BAD_REMOTE_ADAPTER	ERROR	
0xc00000c6	SMB_STATUS_PRINT_QUEUE_FULL		V
0xc00000c7	SMB_STATUS_NO_SPOOL_SPACE		V
0xc00000c8	SMB_STATUS_PRINT_CANCELLED		V
0xc00000c9	SMB_STATUS_NETWORK_NAME_DELETED		V
0xc00000ca	SMB_STATUS_NETWORK_ACCESS_DENIED		V
0xc00000cb	SMB_STATUS_BAD_DEVICE_TYPE		V
0xc00000cc	SMB_STATUS_BAD_NETWORK_NAME		V
0xc00000cd	NT_STATUS_TOO_MANY_NAMES	ERROR	
0xc00000ce	SMB_STATUS_TOO_MANY_SESSIONS		V
0xc00000cf	NT_STATUS_SHARING_PAUSED	ERROR	
0xc00000d0	SMB_STATUS_REQUEST_NOT_ACCEPTED		V
0xc00000d1	NT_STATUS_REDIRECTOR_PAUSED	ERROR	
0xc00000d2	NT_STATUS_NET_WRITE_FAULT	ERROR	
0xc00000d3	NT_STATUS_PROFILING_AT_LIMIT	ERROR	
0xc00000d4	SMB_STATUS_NOT_SAME_DEVICE		V
0xc00000d5	SMB_STATUS_FILE_RENAMED		V
0xc00000d6	NT_STATUS_VIRTUAL_CIRCUIT_CLOSED	ERROR	
0xc00000d7	NT_STATUS_NO_SECURITY_ON_OBJECT	ERROR	
0xc00000d8	NT_STATUS_CANT_WAIT	ERROR	
0xc00000d9	SMB_STATUS_PIPE_EMPTY		V
0xc00000da	NT_STATUS_CANT_ACCESS_DOMAIN_INFO	ERROR	
0xc00000db	NT_STATUS_CANT_TERMINATE_SELF	ERROR	
0xc00000dc	NT_STATUS_INVALID_SERVER_STATE	ERROR	
0xc00000dd	NT_STATUS_INVALID_DOMAIN_STATE	ERROR	
0xc00000de	NT_STATUS_INVALID_DOMAIN_ROLE	ERROR	
0xc00000df	NT_STATUS_NO_SUCH_DOMAIN	ERROR	
0xc00000e0	NT_STATUS_DOMAIN_EXISTS	ERROR	
0xc00000e1	NT_STATUS_DOMAIN_LIMIT_EXCEEDED	ERROR	
0xc00000e2	NT_STATUS_OPLOCK_NOT_GRANTED	ERROR	
0xc00000e3	NT_STATUS_INVALID_OPLOCK_PROTOCOL	ERROR	
0xc00000e4	NT_STATUS_INTERNAL_DB_CORRUPTION	ERROR	
0xc00000e5	NT_STATUS_INTERNAL_ERROR	ERROR	
0xc00000e6	NT_STATUS_GENERIC_NOT_MAPPED	ERROR	
0xc00000e7	NT_STATUS_BAD_DESCRIPTOR_FORMAT	ERROR	
0xc00000e8	NT_STATUS_INVALID_USER_BUFFER	ERROR	

Co

Table 13.10 – continued from previous page

Code	Status	NTSTATUS severity	S
0xc00000e9	NT_STATUS_UNEXPECTED_IO_ERROR	ERROR	
0xc00000ea	NT_STATUS_UNEXPECTED_MM_CREATE_ERR	ERROR	
0xc00000eb	NT_STATUS_UNEXPECTED_MM_MAP_ERROR	ERROR	
0xc00000ec	NT_STATUS_UNEXPECTED_MM_EXTEND_ERR	ERROR	
0xc00000ed	NT_STATUS_NOT_LOGON_PROCESS	ERROR	
0xc00000ee	NT_STATUS_LOGON_SESSION_EXISTS	ERROR	
0xc00000ef	NT_STATUS_INVALID_PARAMETER_1	ERROR	
0xc00000f0	NT_STATUS_INVALID_PARAMETER_2	ERROR	
0xc00000f1	NT_STATUS_INVALID_PARAMETER_3	ERROR	
0xc00000f2	NT_STATUS_INVALID_PARAMETER_4	ERROR	
0xc00000f3	NT_STATUS_INVALID_PARAMETER_5	ERROR	
0xc00000f4	NT_STATUS_INVALID_PARAMETER_6	ERROR	
0xc00000f5	NT_STATUS_INVALID_PARAMETER_7	ERROR	
0xc00000f6	NT_STATUS_INVALID_PARAMETER_8	ERROR	
0xc00000f7	NT_STATUS_INVALID_PARAMETER_9	ERROR	
0xc00000f8	NT_STATUS_INVALID_PARAMETER_10	ERROR	
0xc00000f9	NT_STATUS_INVALID_PARAMETER_11	ERROR	
0xc00000fa	NT_STATUS_INVALID_PARAMETER_12	ERROR	
0xc00000fb	SMB_STATUS_REDIRECTOR_NOT_STARTED		W
0xc00000fc	NT_STATUS_REDIRECTOR_STARTED	ERROR	
0xc00000fd	NT_STATUS_STACK_OVERFLOW	ERROR	
0xc00000fe	NT_STATUS_NO_SUCH_PACKAGE	ERROR	
0xc00000ff	NT_STATUS_BAD_FUNCTION_TABLE	ERROR	
0xc0000100	NT_STATUS_VARIABLE_NOT_FOUND	ERROR	
0xc0000101	SMB_STATUS_DIRECTORY_NOT_EMPTY		W
0xc0000102	NT_STATUS_FILE_CORRUPT_ERROR	ERROR	
0xc0000103	NT_STATUS_NOT_A_DIRECTORY	ERROR	
0xc0000104	NT_STATUS_BAD_LOGON_SESSION_STATE	ERROR	
0xc0000105	NT_STATUS_LOGON_SESSION_COLLISION	ERROR	
0xc0000106	NT_STATUS_NAME_TOO_LONG	ERROR	
0xc0000107	NT_STATUS_FILES_OPEN	ERROR	
0xc0000108	NT_STATUS_CONNECTION_IN_USE	ERROR	
0xc0000109	NT_STATUS_MESSAGE_NOT_FOUND	ERROR	
0xc000010a	SMB_STATUS_PROCESS_IS_TERMINATING		W
0xc000010b	NT_STATUS_INVALID_LOGON_TYPE	ERROR	
0xc000010c	NT_STATUS_NO_GUID_TRANSLATION	ERROR	
0xc000010d	NT_STATUS_CANNOT_IMPERSONATE	ERROR	
0xc000010e	NT_STATUS_IMAGE_ALREADY_LOADED	ERROR	
0xc000010f	NT_STATUS_ABIOS_NOT_PRESENT	ERROR	
0xc0000110	NT_STATUS_ABIOS_LID_NOT_EXIST	ERROR	
0xc0000111	NT_STATUS_ABIOS_LID_ALREADY_OWNED	ERROR	
0xc0000112	NT_STATUS_ABIOS_NOT_LID_OWNER	ERROR	
0xc0000113	NT_STATUS_ABIOS_INVALID_COMMAND	ERROR	
0xc0000114	NT_STATUS_ABIOS_INVALID_LID	ERROR	
0xc0000115	NT_STATUS_ABIOS_SELECTOR_NOT_AVAILABLE	ERROR	
0xc0000116	NT_STATUS_ABIOS_INVALID_SELECTOR	ERROR	
0xc0000117	NT_STATUS_NO_LDT	ERROR	
0xc0000118	NT_STATUS_INVALID_LDT_SIZE	ERROR	
0xc0000119	NT_STATUS_INVALID_LDT_OFFSET	ERROR	
0xc000011a	NT_STATUS_INVALID_LDT_DESCRIPTOR	ERROR	
0xc000011b	NT_STATUS_INVALID_IMAGE_NE_FORMAT	ERROR	
0xc000011c	NT_STATUS_RXACT_INVALID_STATE	ERROR	
0xc000011d	NT_STATUS_RXACT_COMMIT_FAILURE	ERROR	
0xc000011e	NT_STATUS_MAPPED_FILE_SIZE_ZERO	ERROR	
0xc000011f	SMB_STATUS_TOO_MANY_OPENED_FILES		W

Co

Table 13.10 – continued from previous page

Code	Status	NTSTATUS severity	S
0xc0000120	NT_STATUS_CANCELLED	ERROR	
0xc0000121	SMB_STATUS_CANNOT_DELETE		W
0xc0000122	NT_STATUS_INVALID_COMPUTER_NAME	ERROR	
0xc0000123	SMB_STATUS_FILE_DELETED		W
0xc0000124	NT_STATUS_SPECIAL_ACCOUNT	ERROR	
0xc0000125	NT_STATUS_SPECIAL_GROUP	ERROR	
0xc0000126	NT_STATUS_SPECIAL_USER	ERROR	
0xc0000127	NT_STATUS_MEMBERS_PRIMARY_GROUP	ERROR	
0xc0000128	SMB_STATUS_FILE_CLOSED		W
0xc0000129	NT_STATUS_TOO_MANY_THREADS	ERROR	
0xc000012a	NT_STATUS_THREAD_NOT_IN_PROCESS	ERROR	
0xc000012b	NT_STATUS_TOKEN_ALREADY_IN_USE	ERROR	
0xc000012c	NT_STATUS_PAGEFILE_QUOTA_EXCEEDED	ERROR	
0xc000012d	NT_STATUS_COMMITMENT_LIMIT	ERROR	
0xc000012e	NT_STATUS_INVALID_IMAGE_LE_FORMAT	ERROR	
0xc000012f	NT_STATUS_INVALID_IMAGE_NOT_MZ	ERROR	
0xc0000130	NT_STATUS_INVALID_IMAGE_PROTECT	ERROR	
0xc0000131	NT_STATUS_INVALID_IMAGE_WIN_16	ERROR	
0xc0000132	NT_STATUS_LOGON_SERVER_CONFLICT	ERROR	
0xc0000133	NT_STATUS_TIME_DIFFERENCE_AT_DC	ERROR	
0xc0000134	NT_STATUS_SYNCHRONIZATION_REQUIRED	ERROR	
0xc0000135	NT_STATUS_DLL_NOT_FOUND	ERROR	
0xc0000136	NT_STATUS_OPEN_FAILED	ERROR	
0xc0000137	NT_STATUS_IO_PRIVILEGE_FAILED	ERROR	
0xc0000138	NT_STATUS_ORDINAL_NOT_FOUND	ERROR	
0xc0000139	NT_STATUS_ENTRYPOINT_NOT_FOUND	ERROR	
0xc000013a	NT_STATUS_CONTROL_C_EXIT	ERROR	
0xc000013b	NT_STATUS_LOCAL_DISCONNECT	ERROR	
0xc000013c	NT_STATUS_REMOTE_DISCONNECT	ERROR	
0xc000013d	NT_STATUS_REMOTE_RESOURCES	ERROR	
0xc000013e	NT_STATUS_LINK_FAILED	ERROR	
0xc000013f	NT_STATUS_LINK_TIMEOUT	ERROR	
0xc0000140	NT_STATUS_INVALID_CONNECTION	ERROR	
0xc0000141	NT_STATUS_INVALID_ADDRESS	ERROR	
0xc0000142	NT_STATUS_DLL_INIT_FAILED	ERROR	
0xc0000143	NT_STATUS_MISSING_SYSTEMFILE	ERROR	
0xc0000144	NT_STATUS_UNHANDLED_EXCEPTION	ERROR	
0xc0000145	NT_STATUS_APP_INIT_FAILURE	ERROR	
0xc0000146	NT_STATUS_PAGEFILE_CREATE_FAILED	ERROR	
0xc0000147	NT_STATUS_NO_PAGEFILE	ERROR	
0xc0000148	NT_STATUS_INVALID_LEVEL	ERROR	
0xc0000149	NT_STATUS_WRONG_PASSWORD_CORE	ERROR	
0xc000014a	NT_STATUS_ILLEGAL_FLOAT_CONTEXT	ERROR	
0xc000014b	NT_STATUS_PIPE_BROKEN	ERROR	
0xc000014c	NT_STATUS_REGISTRY_CORRUPT	ERROR	
0xc000014d	NT_STATUS_REGISTRY_IO_FAILED	ERROR	
0xc000014e	NT_STATUS_NO_EVENT_PAIR	ERROR	
0xc000014f	NT_STATUS_UNRECOGNIZED_VOLUME	ERROR	
0xc0000150	NT_STATUS_SERIAL_NO_DEVICE_INITED	ERROR	
0xc0000151	NT_STATUS_NO_SUCH_ALIAS	ERROR	
0xc0000152	NT_STATUS_MEMBER_NOT_IN_ALIAS	ERROR	
0xc0000153	NT_STATUS_MEMBER_IN_ALIAS	ERROR	
0xc0000154	NT_STATUS_ALIAS_EXISTS	ERROR	
0xc0000155	NT_STATUS_LOGON_NOT_GRANTED	ERROR	
0xc0000156	NT_STATUS_TOO_MANY_SECRETS	ERROR	

Co

Table 13.10 – continued from previous page

Code	Status	NTSTATUS severity	S
0xc0000157	NT_STATUS_SECRET_TOO_LONG	ERROR	
0xc0000158	NT_STATUS_INTERNAL_DB_ERROR	ERROR	
0xc0000159	NT_STATUS_FULLSCREEN_MODE	ERROR	
0xc000015a	NT_STATUS_TOO_MANY_CONTEXT_IDS	ERROR	
0xc000015b	NT_STATUS_LOGON_TYPE_NOT_GRANTED	ERROR	
0xc000015c	NT_STATUS_NOT_REGISTRY_FILE	ERROR	
0xc000015d	NT_STATUS_NT_CROSS_ENCRYPTION_REQUIRED	ERROR	
0xc000015e	NT_STATUS_DOMAIN_CTRLR_CONFIG_ERROR	ERROR	
0xc000015f	NT_STATUS_FT_MISSING_MEMBER	ERROR	
0xc0000160	NT_STATUS_ILL_FORMED_SERVICE_ENTRY	ERROR	
0xc0000161	NT_STATUS_ILLEGAL_CHARACTER	ERROR	
0xc0000162	NT_STATUS_UNMAPPABLE_CHARACTER	ERROR	
0xc0000163	NT_STATUS_UNDEFINED_CHARACTER	ERROR	
0xc0000164	NT_STATUS_FLOPPY_VOLUME	ERROR	
0xc0000165	NT_STATUS_FLOPPY_ID_MARK_NOT_FOUND	ERROR	
0xc0000166	NT_STATUS_FLOPPY_WRONG_CYLINDER	ERROR	
0xc0000167	NT_STATUS_FLOPPY_UNKNOWN_ERROR	ERROR	
0xc0000168	NT_STATUS_FLOPPY_BAD_REGISTERS	ERROR	
0xc0000169	NT_STATUS_DISK_RECALIBRATE_FAILED	ERROR	
0xc000016a	NT_STATUS_DISK_OPERATION_FAILED	ERROR	
0xc000016b	NT_STATUS_DISK_RESET_FAILED	ERROR	
0xc000016c	NT_STATUS_SHARED_IRQ_BUSY	ERROR	
0xc000016d	NT_STATUS_FT_ORPHANING	ERROR	
0xc000016e	NT_STATUS_BIOS_FAILED_TO_CONNECT_INTERRUPT	ERROR	
0xc0000172	NT_STATUS_PARTITION_FAILURE	ERROR	
0xc0000173	NT_STATUS_INVALID_BLOCK_LENGTH	ERROR	
0xc0000174	NT_STATUS_DEVICE_NOT_PARTITIONED	ERROR	
0xc0000175	NT_STATUS_UNABLE_TO_LOCK_MEDIA	ERROR	
0xc0000176	NT_STATUS_UNABLE_TO_UNLOAD_MEDIA	ERROR	
0xc0000177	NT_STATUS_EOM_OVERFLOW	ERROR	
0xc0000178	NT_STATUS_NO_MEDIA	ERROR	
0xc000017a	NT_STATUS_NO_SUCH_MEMBER	ERROR	
0xc000017b	NT_STATUS_INVALID_MEMBER	ERROR	
0xc000017c	NT_STATUS_KEY_DELETED	ERROR	
0xc000017d	NT_STATUS_NO_LOG_SPACE	ERROR	
0xc000017e	NT_STATUS_TOO_MANY_SIDS	ERROR	
0xc000017f	NT_STATUS_LM_CROSS_ENCRYPTION_REQUIRED	ERROR	
0xc0000180	NT_STATUS_KEY_HAS_CHILDREN	ERROR	
0xc0000181	NT_STATUS_CHILD_MUST_BE_VOLATILE	ERROR	
0xc0000182	NT_STATUS_DEVICE_CONFIGURATION_ERROR	ERROR	
0xc0000183	NT_STATUS_DRIVER_INTERNAL_ERROR	ERROR	
0xc0000184	SMB_STATUS_INVALID_DEVICE_STATE		E
0xc0000185	NT_STATUS_IO_DEVICE_ERROR	ERROR	
0xc0000186	NT_STATUS_DEVICE_PROTOCOL_ERROR	ERROR	
0xc0000187	NT_STATUS_BACKUP_CONTROLLER	ERROR	
0xc0000188	NT_STATUS_LOG_FILE_FULL	ERROR	
0xc0000189	NT_STATUS_TOO_LATE	ERROR	
0xc000018a	NT_STATUS_NO_TRUST_LSA_SECRET	ERROR	
0xc000018b	NT_STATUS_NO_TRUST_SAM_ACCOUNT	ERROR	
0xc000018c	NT_STATUS_TRUSTED_DOMAIN_FAILURE	ERROR	
0xc000018d	NT_STATUS_TRUSTED_RELATIONSHIP_FAILURE	ERROR	
0xc000018e	NT_STATUS_EVENTLOG_FILE_CORRUPT	ERROR	
0xc000018f	NT_STATUS_EVENTLOG_CANT_START	ERROR	
0xc0000190	NT_STATUS_TRUST_FAILURE	ERROR	
0xc0000191	NT_STATUS_MUTANT_LIMIT_EXCEEDED	ERROR	

Co

Table 13.10 – continued from previous page

Code	Status	NTSTATUS severity	S
0xc0000192	NT_STATUS_NETLOGON_NOT_STARTED	ERROR	
0xc0000193	SMB_STATUS_ACCOUNT_EXPIRED		W
0xc0000194	NT_STATUS_POSSIBLE_DEADLOCK	ERROR	
0xc0000195	NT_STATUS_NETWORK_CREDENTIAL_CONFLICT	ERROR	
0xc0000196	NT_STATUS_REMOTE_SESSION_LIMIT	ERROR	
0xc0000197	NT_STATUS_EVENTLOG_FILE_CHANGED	ERROR	
0xc0000198	NT_STATUS_NOLOGON_INTERDOMAIN_TRUST_ACCOUNT	ERROR	
0xc0000199	NT_STATUS_NOLOGON_WORKSTATION_TRUST_ACCOUNT	ERROR	
0xc000019a	NT_STATUS_NOLOGON_SERVER_TRUST_ACCOUNT	ERROR	
0xc000019b	NT_STATUS_DOMAIN_TRUST_INCONSISTENT	ERROR	
0xc000019c	NT_STATUS_FS_DRIVER_REQUIRED	ERROR	
0xc0000202	NT_STATUS_NO_USER_SESSION_KEY	ERROR	
0xc0000203	NT_STATUS_USER_SESSION_DELETED	ERROR	
0xc0000204	NT_STATUS_RESOURCE_LANG_NOT_FOUND	ERROR	
0xc0000205	SMB_STATUS_INSUFF_SERVER_RESOURCES		W
0xc0000206	NT_STATUS_INVALID_BUFFER_SIZE	ERROR	
0xc0000207	NT_STATUS_INVALID_ADDRESS_COMPONENT	ERROR	
0xc0000208	NT_STATUS_INVALID_ADDRESS_WILDCARD	ERROR	
0xc0000209	NT_STATUS_TOO_MANY_ADDRESSES	ERROR	
0xc000020a	NT_STATUS_ADDRESS_ALREADY_EXISTS	ERROR	
0xc000020b	NT_STATUS_ADDRESS_CLOSED	ERROR	
0xc000020c	NT_STATUS_CONNECTION_DISCONNECTED	ERROR	
0xc000020d	NT_STATUS_CONNECTION_RESET	ERROR	
0xc000020e	NT_STATUS_TOO_MANY_NODES	ERROR	
0xc000020f	NT_STATUS_TRANSACTION_ABORTED	ERROR	
0xc0000210	NT_STATUS_TRANSACTION_TIMED_OUT	ERROR	
0xc0000211	NT_STATUS_TRANSACTION_NO_RELEASE	ERROR	
0xc0000212	NT_STATUS_TRANSACTION_NO_MATCH	ERROR	
0xc0000213	NT_STATUS_TRANSACTION_RESPONDED	ERROR	
0xc0000214	NT_STATUS_TRANSACTION_INVALID_ID	ERROR	
0xc0000215	NT_STATUS_TRANSACTION_INVALID_TYPE	ERROR	
0xc0000216	NT_STATUS_NOT_SERVER_SESSION	ERROR	
0xc0000217	NT_STATUS_NOT_CLIENT_SESSION	ERROR	
0xc0000218	NT_STATUS_CANNOT_LOAD_REGISTRY_FILE	ERROR	
0xc0000219	NT_STATUS_DEBUG_ATTACH_FAILED	ERROR	
0xc000021a	NT_STATUS_SYSTEM_PROCESS_TERMINATED	ERROR	
0xc000021b	NT_STATUS_DATA_NOT_ACCEPTED	ERROR	
0xc000021c	NT_STATUS_NO_BROWSER_SERVERS_FOUND	ERROR	
0xc000021d	NT_STATUS_VDM_HARD_ERROR	ERROR	
0xc000021e	NT_STATUS_DRIVER_CANCEL_TIMEOUT	ERROR	
0xc000021f	NT_STATUS_REPLY_MESSAGE_MISMATCH	ERROR	
0xc0000220	NT_STATUS_MAPPED_ALIGNMENT	ERROR	
0xc0000221	NT_STATUS_IMAGE_CHECKSUM_MISMATCH	ERROR	
0xc0000222	NT_STATUS_LOST_WRITEBEHIND_DATA	ERROR	
0xc0000223	NT_STATUS_CLIENT_SERVER_PARAMETERS_INVALID	ERROR	
0xc0000224	SMB_STATUS_PASSWORD_MUST_CHANGE		W
0xc0000225	NT_STATUS_NOT_FOUND	ERROR	
0xc0000226	NT_STATUS_NOT_TINY_STREAM	ERROR	
0xc0000227	NT_STATUS_RECOVERY_FAILURE	ERROR	
0xc0000228	NT_STATUS_STACK_OVERFLOW_READ	ERROR	
0xc0000229	NT_STATUS_FAIL_CHECK	ERROR	
0xc000022a	NT_STATUS_DUPLICATE_OBJECTID	ERROR	
0xc000022b	NT_STATUS_OBJECTID_EXISTS	ERROR	
0xc000022c	NT_STATUS_CONVERT_TO_LARGE	ERROR	
0xc000022d	NT_STATUS_RETRY	ERROR	

Co

Table 13.10 – continued from previous page

Code	Status	NTSTATUS severity	S
0xc000022e	NT_STATUS_FOUND_OUT_OF_SCOPE	ERROR	
0xc000022f	NT_STATUS_ALLOCATE_BUCKET	ERROR	
0xc0000230	NT_STATUS_PROPSET_NOT_FOUND	ERROR	
0xc0000231	NT_STATUS_MARSHALL_OVERFLOW	ERROR	
0xc0000232	NT_STATUS_INVALID_VARIANT	ERROR	
0xc0000233	NT_STATUS_DOMAIN_CONTROLLER_NOT_FOUND	ERROR	
0xc0000234	NT_STATUS_ACCOUNT_LOCKED_OUT	ERROR	
0xc0000235	SMB_STATUS_HANDLE_NOT_CLOSABLE		V
0xc0000236	NT_STATUS_CONNECTION_REFUSED	ERROR	
0xc0000237	NT_STATUS_GRACEFUL_DISCONNECT	ERROR	
0xc0000238	NT_STATUS_ADDRESS_ALREADY_ASSOCIATED	ERROR	
0xc0000239	NT_STATUS_ADDRESS_NOT_ASSOCIATED	ERROR	
0xc000023a	NT_STATUS_CONNECTION_INVALID	ERROR	
0xc000023b	NT_STATUS_CONNECTION_ACTIVE	ERROR	
0xc000023c	NT_STATUS_NETWORK_UNREACHABLE	ERROR	
0xc000023d	NT_STATUS_HOST_UNREACHABLE	ERROR	
0xc000023e	NT_STATUS_PROTOCOL_UNREACHABLE	ERROR	
0xc000023f	NT_STATUS_PORT_UNREACHABLE	ERROR	
0xc0000240	NT_STATUS_REQUEST_ABORTED	ERROR	
0xc0000241	NT_STATUS_CONNECTION_ABORTED	ERROR	
0xc0000242	NT_STATUS_BAD_COMPRESSION_BUFFER	ERROR	
0xc0000243	NT_STATUS_USER_MAPPED_FILE	ERROR	
0xc0000244	NT_STATUS_AUDIT_FAILED	ERROR	
0xc0000245	NT_STATUS_TIMER_RESOLUTION_NOT_SET	ERROR	
0xc0000246	NT_STATUS_CONNECTION_COUNT_LIMIT	ERROR	
0xc0000247	NT_STATUS_LOGIN_TIME_RESTRICTION	ERROR	
0xc0000248	NT_STATUS_LOGIN_WKSTA_RESTRICTION	ERROR	
0xc0000249	NT_STATUS_IMAGE_MP_UP_MISMATCH	ERROR	
0xc0000250	NT_STATUS_INSUFFICIENT_LOGON_INFO	ERROR	
0xc0000251	NT_STATUS_BAD_DLL_ENTRYPOINT	ERROR	
0xc0000252	NT_STATUS_BAD_SERVICE_ENTRYPOINT	ERROR	
0xc0000253	NT_STATUS_LPC_REPLY_LOST	ERROR	
0xc0000254	NT_STATUS_IP_ADDRESS_CONFLICT1	ERROR	
0xc0000255	NT_STATUS_IP_ADDRESS_CONFLICT2	ERROR	
0xc0000256	NT_STATUS_REGISTRY_QUOTA_LIMIT	ERROR	
0xc0000257	SMB_STATUS_PATH_NOT_COVERED		V
0xc0000258	NT_STATUS_NO_CALLBACK_ACTIVE	ERROR	
0xc0000259	NT_STATUS_LICENSE_QUOTA_EXCEEDED	ERROR	
0xc000025a	NT_STATUS_PWD_TOO_SHORT	ERROR	
0xc000025b	NT_STATUS_PWD_TOO_RECENT	ERROR	
0xc000025c	NT_STATUS_PWD_HISTORY_CONFLICT	ERROR	
0xc000025e	NT_STATUS_PLUGPLAY_NO_DEVICE	ERROR	
0xc000025f	NT_STATUS_UNSUPPORTED_COMPRESSION	ERROR	
0xc0000260	NT_STATUS_INVALID_HW_PROFILE	ERROR	
0xc0000261	NT_STATUS_INVALID_PLUGPLAY_DEVICE_PATH	ERROR	
0xc0000262	NT_STATUS_DRIVER_ORDINAL_NOT_FOUND	ERROR	
0xc0000263	NT_STATUS_DRIVER_ENTRYPOINT_NOT_FOUND	ERROR	
0xc0000264	NT_STATUS_RESOURCE_NOT_OWNED	ERROR	
0xc0000265	NT_STATUS_TOO_MANY_LINKS	ERROR	
0xc0000266	NT_STATUS_QUOTA_LIST_INCONSISTENT	ERROR	
0xc0000267	NT_STATUS_FILE_IS_OFFLINE	ERROR	
0xc0000268	NT_STATUS_EVALUATION_EXPIRATION	ERROR	
0xc0000269	NT_STATUS_ILLEGAL_DLL_RELOCATION	ERROR	
0xc000026a	NT_STATUS_LICENSE_VIOLATION	ERROR	
0xc000026b	NT_STATUS_DLL_INIT_FAILED_LOGOFF	ERROR	

Co

Table 13.10 – continued from previous page

Code	Status	NTSTATUS severity	S
0xc000026c	NT_STATUS_DRIVER_UNABLE_TO_LOAD	ERROR	
0xc000026d	NT_STATUS_DFS_UNAVAILABLE	ERROR	
0xc000026e	NT_STATUS_VOLUME_DISMOUNTED	ERROR	
0xc000026f	NT_STATUS_WX86_INTERNAL_ERROR	ERROR	
0xc0000270	NT_STATUS_WX86_FLOAT_STACK_CHECK	ERROR	
0xc0000271	NT_STATUS_VALIDATE_CONTINUE	ERROR	
0xc0000272	NT_STATUS_NO_MATCH	ERROR	
0xc0000273	NT_STATUS_NO_MORE_MATCHES	ERROR	
0xc0000275	NT_STATUS_NOT_A_REPARSE_POINT	ERROR	
0xc0000276	NT_STATUS_IO_REPARSE_TAG_INVALID	ERROR	
0xc0000277	NT_STATUS_IO_REPARSE_TAG_MISMATCH	ERROR	
0xc0000278	NT_STATUS_IO_REPARSE_DATA_INVALID	ERROR	
0xc0000279	NT_STATUS_IO_REPARSE_TAG_NOT_HANDLED	ERROR	
0xc0000280	NT_STATUS_REPARSE_POINT_NOT_RESOLVED	ERROR	
0xc0000281	NT_STATUS_DIRECTORY_IS_A_REPARSE_POINT	ERROR	
0xc0000282	NT_STATUS_RANGE_LIST_CONFLICT	ERROR	
0xc0000283	NT_STATUS_SOURCE_ELEMENT_EMPTY	ERROR	
0xc0000284	NT_STATUS_DESTINATION_ELEMENT_FULL	ERROR	
0xc0000285	NT_STATUS_ILLEGAL_ELEMENT_ADDRESS	ERROR	
0xc0000286	NT_STATUS_MAGAZINE_NOT_PRESENT	ERROR	
0xc0000287	NT_STATUS_REINITIALIZATION_NEEDED	ERROR	
0xc000028a	NT_STATUS_ENCRYPTION_FAILED	ERROR	
0xc000028b	NT_STATUS_DECRYPTION_FAILED	ERROR	
0xc000028c	NT_STATUS_RANGE_NOT_FOUND	ERROR	
0xc000028d	NT_STATUS_NO_RECOVERY_POLICY	ERROR	
0xc000028e	NT_STATUS_NO_EFS	ERROR	
0xc000028f	NT_STATUS_WRONG_EFS	ERROR	
0xc0000290	NT_STATUS_NO_USER_KEYS	ERROR	
0xc0000291	NT_STATUS_FILE_NOT_ENCRYPTED	ERROR	
0xc0000292	NT_STATUS_NOT_EXPORT_FORMAT	ERROR	
0xc0000293	NT_STATUS_FILE_ENCRYPTED	ERROR	
0xc0000295	NT_STATUS_WMI_GUID_NOT_FOUND	ERROR	
0xc0000296	NT_STATUS_WMI_INSTANCE_NOT_FOUND	ERROR	
0xc0000297	NT_STATUS_WMI_ITEMID_NOT_FOUND	ERROR	
0xc0000298	NT_STATUS_WMI_TRY_AGAIN	ERROR	
0xc0000299	NT_STATUS_SHARED_POLICY	ERROR	
0xc000029a	NT_STATUS_POLICY_OBJECT_NOT_FOUND	ERROR	
0xc000029b	NT_STATUS_POLICY_ONLY_IN_DS	ERROR	
0xc000029c	NT_STATUS_VOLUME_NOT_UPGRADED	ERROR	
0xc000029d	NT_STATUS_REMOTE_STORAGE_NOT_ACTIVE	ERROR	
0xc000029e	NT_STATUS_REMOTE_STORAGE_MEDIA_ERROR	ERROR	
0xc000029f	NT_STATUS_NO_TRACKING_SERVICE	ERROR	
0xc00002a0	NT_STATUS_SERVER_SID_MISMATCH	ERROR	
0xc00002a1	NT_STATUS_DS_NO_ATTRIBUTE_OR_VALUE	ERROR	
0xc00002a2	NT_STATUS_DS_INVALID_ATTRIBUTE_SYNTAX	ERROR	
0xc00002a3	NT_STATUS_DS_ATTRIBUTE_TYPE_UNDEFINED	ERROR	
0xc00002a4	NT_STATUS_DS_ATTRIBUTE_OR_VALUE_EXISTS	ERROR	
0xc00002a5	NT_STATUS_DS_BUSY	ERROR	
0xc00002a6	NT_STATUS_DS_UNAVAILABLE	ERROR	
0xc00002a7	NT_STATUS_DS_NO_RIDS_ALLOCATED	ERROR	
0xc00002a8	NT_STATUS_DS_NO_MORE_RIDS	ERROR	
0xc00002a9	NT_STATUS_DS_INCORRECT_ROLE_OWNER	ERROR	
0xc00002aa	NT_STATUS_DS_RIDMGR_INIT_ERROR	ERROR	
0xc00002ab	NT_STATUS_DS_OBJ_CLASS_VIOLATION	ERROR	
0xc00002ac	NT_STATUS_DS_CANT_ON_NON_LEAF	ERROR	

Co

Table 13.10 – continued from previous page

Code	Status	NTSTATUS severity	S
0xc00002ad	NT_STATUS_DS_CANT_ON_RDN	ERROR	
0xc00002ae	NT_STATUS_DS_CANT_MOD_OBJ_CLASS	ERROR	
0xc00002af	NT_STATUS_DS_CROSS_DOM_MOVE_FAILED	ERROR	
0xc00002b0	NT_STATUS_DS_GC_NOT_AVAILABLE	ERROR	
0xc00002b1	NT_STATUS_DIRECTORY_SERVICE_REQUIRED	ERROR	
0xc00002b2	NT_STATUS_REPARSE_ATTRIBUTE_CONFLICT	ERROR	
0xc00002b3	NT_STATUS_CANT_ENABLE_DENY_ONLY	ERROR	
0xc00002b4	NT_STATUS_FLOAT_MULTIPLE_FAULTS	ERROR	
0xc00002b5	NT_STATUS_FLOAT_MULTIPLE_TRAPS	ERROR	
0xc00002b6	NT_STATUS_DEVICE_REMOVED	ERROR	
0xc00002b7	NT_STATUS_JOURNAL_DELETE_IN_PROGRESS	ERROR	
0xc00002b8	NT_STATUS_JOURNAL_NOT_ACTIVE	ERROR	
0xc00002b9	NT_STATUS_NOINTERFACE	ERROR	
0xc00002c1	NT_STATUS_DS_ADMIN_LIMIT_EXCEEDED	ERROR	
0xc00002c2	NT_STATUS_DRIVER_FAILED_SLEEP	ERROR	
0xc00002c3	NT_STATUS_MUTUAL_AUTHENTICATION_FAILED	ERROR	
0xc00002c4	NT_STATUS_CORRUPT_SYSTEM_FILE	ERROR	
0xc00002c5	NT_STATUS_DATATYPE_MISALIGNMENT_ERROR	ERROR	
0xc00002c6	NT_STATUS_WMI_READ_ONLY	ERROR	
0xc00002c7	NT_STATUS_WMI_SET_FAILURE	ERROR	
0xc00002c8	NT_STATUS_COMMITMENT_MINIMUM	ERROR	
0xc00002c9	NT_STATUS_REG_NAT_CONSUMPTION	ERROR	
0xc00002ca	NT_STATUS_TRANSPORT_FULL	ERROR	
0xc00002cb	NT_STATUS_DS_SAM_INIT_FAILURE	ERROR	
0xc00002cc	NT_STATUS_ONLY_IF_CONNECTED	ERROR	
0xc00002cd	NT_STATUS_DS_SENSITIVE_GROUP_VIOLATION	ERROR	
0xc00002ce	NT_STATUS_PNP_RESTART_ENUMERATION	ERROR	
0xc00002cf	NT_STATUS_JOURNAL_ENTRY_DELETED	ERROR	
0xc00002d0	NT_STATUS_DS_CANT_MOD_PRIMARYGROUPID	ERROR	
0xc00002d1	NT_STATUS_SYSTEM_IMAGE_BAD_SIGNATURE	ERROR	
0xc00002d2	NT_STATUS_PNP_REBOOT_REQUIRED	ERROR	
0xc00002d3	NT_STATUS_POWER_STATE_INVALID	ERROR	
0xc00002d4	NT_STATUS_DS_INVALID_GROUP_TYPE	ERROR	
0xc00002d5	NT_STATUS_DS_NO_NEST_GLOBALGROUP_IN_MIXEDDOMAIN	ERROR	
0xc00002d6	NT_STATUS_DS_NO_NEST_LOCALGROUP_IN_MIXEDDOMAIN	ERROR	
0xc00002d7	NT_STATUS_DS_GLOBAL_CANT_HAVE_LOCAL_MEMBER	ERROR	
0xc00002d8	NT_STATUS_DS_GLOBAL_CANT_HAVE_UNIVERSAL_MEMBER	ERROR	
0xc00002d9	NT_STATUS_DS_UNIVERSAL_CANT_HAVE_LOCAL_MEMBER	ERROR	
0xc00002da	NT_STATUS_DS_GLOBAL_CANT_HAVE_CROSSDOMAIN_MEMBER	ERROR	
0xc00002db	NT_STATUS_DS_LOCAL_CANT_HAVE_CROSSDOMAIN_LOCAL_MEMBER	ERROR	
0xc00002dc	NT_STATUS_DS_HAVE_PRIMARY_MEMBERS	ERROR	
0xc00002dd	NT_STATUS_WMI_NOT_SUPPORTED	ERROR	
0xc00002de	NT_STATUS_INSUFFICIENT_POWER	ERROR	
0xc00002df	NT_STATUS_SAM_NEED_BOOTKEY_PASSWORD	ERROR	
0xc00002e0	NT_STATUS_SAM_NEED_BOOTKEY_FLOPPY	ERROR	
0xc00002e1	NT_STATUS_DS_CANT_START	ERROR	
0xc00002e2	NT_STATUS_DS_INIT_FAILURE	ERROR	
0xc00002e3	NT_STATUS_SAM_INIT_FAILURE	ERROR	
0xc00002e4	NT_STATUS_DS_GC_REQUIRED	ERROR	
0xc00002e5	NT_STATUS_DS_LOCAL_MEMBER_OF_LOCAL_ONLY	ERROR	
0xc00002e6	NT_STATUS_DS_NO_FPO_IN_UNIVERSAL_GROUPS	ERROR	
0xc00002e7	NT_STATUS_DS_MACHINE_ACCOUNT_QUOTA_EXCEEDED	ERROR	
0xc00002e8	NT_STATUS_MULTIPLE_FAULT_VIOLATION	ERROR	
0xc00002e9	NT_STATUS_CURRENT_DOMAIN_NOT_ALLOWED	ERROR	
0xc00002ea	NT_STATUS_CANNOT_MAKE	ERROR	

Co

Table 13.10 – continued from previous page

Code	Status	NTSTATUS severity	S
0xc00002eb	NT_STATUS_SYSTEM_SHUTDOWN	ERROR	
0xc00002ec	NT_STATUS_DS_INIT_FAILURE_CONSOLE	ERROR	
0xc00002ed	NT_STATUS_DS_SAM_INIT_FAILURE_CONSOLE	ERROR	
0xc00002ee	NT_STATUS_UNFINISHED_CONTEXT_DELETED	ERROR	
0xc00002ef	NT_STATUS_NO_TGT_REPLY	ERROR	
0xc00002f0	NT_STATUS_OBJECTID_NOT_FOUND	ERROR	
0xc00002f1	NT_STATUS_NO_IP_ADDRESSES	ERROR	
0xc00002f2	NT_STATUS_WRONG_CREDENTIAL_HANDLE	ERROR	
0xc00002f3	NT_STATUS_CRYPTO_SYSTEM_INVALID	ERROR	
0xc00002f4	NT_STATUS_MAX_REFERRALS_EXCEEDED	ERROR	
0xc00002f5	NT_STATUS_MUST_BE_KDC	ERROR	
0xc00002f6	NT_STATUS_STRONG_CRYPTO_NOT_SUPPORTED	ERROR	
0xc00002f7	NT_STATUS_TOO_MANY_PRINCIPALS	ERROR	
0xc00002f8	NT_STATUS_NO_PA_DATA	ERROR	
0xc00002f9	NT_STATUS_PKINIT_NAME_MISMATCH	ERROR	
0xc00002fa	NT_STATUS_SMARTCARD_LOGON_REQUIRED	ERROR	
0xc00002fb	NT_STATUS_KDC_INVALID_REQUEST	ERROR	
0xc00002fc	NT_STATUS_KDC_UNABLE_TO_REFER	ERROR	
0xc00002fd	NT_STATUS_KDC_UNKNOWN_ETYPE	ERROR	
0xc00002fe	NT_STATUS_SHUTDOWN_IN_PROGRESS	ERROR	
0xc00002ff	NT_STATUS_SERVER_SHUTDOWN_IN_PROGRESS	ERROR	
0xc0000300	NT_STATUS_NOT_SUPPORTED_ON_SBS	ERROR	
0xc0000301	NT_STATUS_WMI_GUID_DISCONNECTED	ERROR	
0xc0000302	NT_STATUS_WMI_ALREADY_DISABLED	ERROR	
0xc0000303	NT_STATUS_WMI_ALREADY_ENABLED	ERROR	
0xc0000304	NT_STATUS_MFT_TOO_FRAGMENTED	ERROR	
0xc0000305	NT_STATUS_COPY_PROTECTION_FAILURE	ERROR	
0xc0000306	NT_STATUS_CSS_AUTHENTICATION_FAILURE	ERROR	
0xc0000307	NT_STATUS_CSS_KEY_NOT_PRESENT	ERROR	
0xc0000308	NT_STATUS_CSS_KEY_NOT_ESTABLISHED	ERROR	
0xc0000309	NT_STATUS_CSS_SCRAMBLED_SECTOR	ERROR	
0xc000030a	NT_STATUS_CSS_REGION_MISMATCH	ERROR	
0xc000030b	NT_STATUS_CSS_RESETS_EXHAUSTED	ERROR	
0xc0000320	NT_STATUS_PKINIT_FAILURE	ERROR	
0xc0000321	NT_STATUS_SMARTCARD_SUBSYSTEM_FAILURE	ERROR	
0xc0000322	NT_STATUS_NO_KERB_KEY	ERROR	
0xc0000350	NT_STATUS_HOST_DOWN	ERROR	
0xc0000351	NT_STATUS_UNSUPPORTED_PREAUTH	ERROR	
0xc0000352	NT_STATUS_EFS_ALG_BLOB_TOO_BIG	ERROR	
0xc0000353	NT_STATUS_PORT_NOT_SET	ERROR	
0xc0000354	NT_STATUS_DEBUGGER_INACTIVE	ERROR	
0xc0000355	NT_STATUS_DS_VERSION_CHECK_FAILURE	ERROR	
0xc0000356	NT_STATUS_AUDITING_DISABLED	ERROR	
0xc0000357	NT_STATUS_PRENT4_MACHINE_ACCOUNT	ERROR	
0xc0000358	NT_STATUS_DS_AG_CANT_HAVE_UNIVERSAL_MEMBER	ERROR	
0xc0000359	NT_STATUS_INVALID_IMAGE_WIN_32	ERROR	
0xc000035a	NT_STATUS_INVALID_IMAGE_WIN_64	ERROR	
0xc000035b	NT_STATUS_BAD_BINDINGS	ERROR	
0xc000035c	NT_STATUS_NETWORK_SESSION_EXPIRED	ERROR	
0xc000035d	NT_STATUS_APPHELP_BLOCK	ERROR	
0xc000035e	NT_STATUS_ALL_SIDS_FILTERED	ERROR	
0xc000035f	NT_STATUS_NOT_SAFE_MODE_DRIVER	ERROR	
0xc0000361	NT_STATUS_ACCESS_DISABLED_BY_POLICY_DEFAULT	ERROR	
0xc0000362	NT_STATUS_ACCESS_DISABLED_BY_POLICY_PATH	ERROR	
0xc0000363	NT_STATUS_ACCESS_DISABLED_BY_POLICY_PUBLISHER	ERROR	

Co

Table 13.10 – continued from previous page

Code	Status	NTSTATUS severity	S
0xc0000364	NT_STATUS_ACCESS_DISABLED_BY_POLICY_OTHER	ERROR	
0xc0000365	NT_STATUS_FAILED_DRIVER_ENTRY	ERROR	
0xc0000366	NT_STATUS_DEVICE_ENUMERATION_ERROR	ERROR	
0xc0000368	NT_STATUS_MOUNT_POINT_NOT_RESOLVED	ERROR	
0xc0000369	NT_STATUS_INVALID_DEVICE_OBJECT_PARAMETER	ERROR	
0xc000036a	NT_STATUS_MCA_OCCURED	ERROR	
0xc000036b	NT_STATUS_DRIVER_BLOCKED_CRITICAL	ERROR	
0xc000036c	NT_STATUS_DRIVER_BLOCKED	ERROR	
0xc000036d	NT_STATUS_DRIVER_DATABASE_ERROR	ERROR	
0xc000036e	NT_STATUS_SYSTEM_HIVE_TOO_LARGE	ERROR	
0xc000036f	NT_STATUS_INVALID_IMPORT_OF_NON_DLL	ERROR	
0xc0000380	NT_STATUS_SMARTCARD_WRONG_PIN	ERROR	
0xc0000381	NT_STATUS_SMARTCARD_CARD_BLOCKED	ERROR	
0xc0000382	NT_STATUS_SMARTCARD_CARD_NOT_AUTHENTICATED	ERROR	
0xc0000383	NT_STATUS_SMARTCARD_NO_CARD	ERROR	
0xc0000384	NT_STATUS_SMARTCARD_NO_KEY_CONTAINER	ERROR	
0xc0000385	NT_STATUS_SMARTCARD_NO_CERTIFICATE	ERROR	
0xc0000386	NT_STATUS_SMARTCARD_NO_KEYSET	ERROR	
0xc0000387	NT_STATUS_SMARTCARD_IO_ERROR	ERROR	
0xc0000388	NT_STATUS_DOWNGRADE_DETECTED	ERROR	
0xc0000389	NT_STATUS_SMARTCARD_CERT_REVOKED	ERROR	
0xc000038a	NT_STATUS_ISSUING_CA_UNTRUSTED	ERROR	
0xc000038b	NT_STATUS_REVOCATION_OFFLINE_C	ERROR	
0xc000038c	NT_STATUS_PKINIT_CLIENT_FAILURE	ERROR	
0xc000038d	NT_STATUS_SMARTCARD_CERT_EXPIRED	ERROR	
0xc000038e	NT_STATUS_DRIVER_FAILED_PRIOR_UNLOAD	ERROR	
0xc0009898	NT_STATUS_WOW_ASSERTION	ERROR	
0xc0020001	RPC_NT_INVALID_STRING_BINDING	ERROR	
0xc0020002	RPC_NT_WRONG_KIND_OF_BINDING	ERROR	
0xc0020003	RPC_NT_INVALID_BINDING	ERROR	
0xc0020004	RPC_NT_PROTSEQ_NOT_SUPPORTED	ERROR	
0xc0020005	RPC_NT_INVALID_RPC_PROTSEQ	ERROR	
0xc0020006	RPC_NT_INVALID_STRING_UUID	ERROR	
0xc0020007	RPC_NT_INVALID_ENDPOINT_FORMAT	ERROR	
0xc0020008	RPC_NT_INVALID_NET_ADDR	ERROR	
0xc0020009	RPC_NT_NO_ENDPOINT_FOUND	ERROR	
0xc002000a	RPC_NT_INVALID_TIMEOUT	ERROR	
0xc002000b	RPC_NT_OBJECT_NOT_FOUND	ERROR	
0xc002000c	RPC_NT_ALREADY_REGISTERED	ERROR	
0xc002000d	RPC_NT_TYPE_ALREADY_REGISTERED	ERROR	
0xc002000e	RPC_NT_ALREADY_LISTENING	ERROR	
0xc002000f	RPC_NT_NO_PROTSEQS_REGISTERED	ERROR	
0xc0020010	RPC_NT_NOT_LISTENING	ERROR	
0xc0020011	RPC_NT_UNKNOWN_MGR_TYPE	ERROR	
0xc0020012	RPC_NT_UNKNOWN_IF	ERROR	
0xc0020013	RPC_NT_NO_BINDINGS	ERROR	
0xc0020014	RPC_NT_NO_PROTSEQS	ERROR	
0xc0020015	RPC_NT_CANT_CREATE_ENDPOINT	ERROR	
0xc0020016	RPC_NT_OUT_OF_RESOURCES	ERROR	
0xc0020017	RPC_NT_SERVER_UNAVAILABLE	ERROR	
0xc0020018	RPC_NT_SERVER_TOO_BUSY	ERROR	
0xc0020019	RPC_NT_INVALID_NETWORK_OPTIONS	ERROR	
0xc002001a	RPC_NT_NO_CALL_ACTIVE	ERROR	
0xc002001b	RPC_NT_CALL_FAILED	ERROR	
0xc002001c	RPC_NT_CALL_FAILED_DNE	ERROR	

Co

Table 13.10 – continued from previous page

Code	Status	NTSTATUS severity	S
0xc002001d	RPC_NT_PROTOCOL_ERROR	ERROR	
0xc002001f	RPC_NT_UNSUPPORTED_TRANS_SYN	ERROR	
0xc0020021	RPC_NT_UNSUPPORTED_TYPE	ERROR	
0xc0020022	RPC_NT_INVALID_TAG	ERROR	
0xc0020023	RPC_NT_INVALID_BOUND	ERROR	
0xc0020024	RPC_NT_NO_ENTRY_NAME	ERROR	
0xc0020025	RPC_NT_INVALID_NAME_SYNTAX	ERROR	
0xc0020026	RPC_NT_UNSUPPORTED_NAME_SYNTAX	ERROR	
0xc0020028	RPC_NT_UUID_NO_ADDRESS	ERROR	
0xc0020029	RPC_NT_DUPLICATE_ENDPOINT	ERROR	
0xc002002a	RPC_NT_UNKNOWN_AUTHN_TYPE	ERROR	
0xc002002b	RPC_NT_MAX_CALLS_TOO_SMALL	ERROR	
0xc002002c	RPC_NT_STRING_TOO_LONG	ERROR	
0xc002002d	RPC_NT_PROTSEQ_NOT_FOUND	ERROR	
0xc002002e	RPC_NT_PROCNUM_OUT_OF_RANGE	ERROR	
0xc002002f	RPC_NT_BINDING_HAS_NO_AUTH	ERROR	
0xc0020030	RPC_NT_UNKNOWN_AUTHN_SERVICE	ERROR	
0xc0020031	RPC_NT_UNKNOWN_AUTHN_LEVEL	ERROR	
0xc0020032	RPC_NT_INVALID_AUTH_IDENTITY	ERROR	
0xc0020033	RPC_NT_UNKNOWN_AUTHZ_SERVICE	ERROR	
0xc0020034	EPT_NT_INVALID_ENTRY	ERROR	
0xc0020035	EPT_NT_CANT_PERFORM_OP	ERROR	
0xc0020036	EPT_NT_NOT_REGISTERED	ERROR	
0xc0020037	RPC_NT_NOTHING_TO_EXPORT	ERROR	
0xc0020038	RPC_NT_INCOMPLETE_NAME	ERROR	
0xc0020039	RPC_NT_INVALID_VERS_OPTION	ERROR	
0xc002003a	RPC_NT_NO_MORE_MEMBERS	ERROR	
0xc002003b	RPC_NT_NOT_ALL_OBJS_UNEXPORTED	ERROR	
0xc002003c	RPC_NT_INTERFACE_NOT_FOUND	ERROR	
0xc002003d	RPC_NT_ENTRY_ALREADY_EXISTS	ERROR	
0xc002003e	RPC_NT_ENTRY_NOT_FOUND	ERROR	
0xc002003f	RPC_NT_NAME_SERVICE_UNAVAILABLE	ERROR	
0xc0020040	RPC_NT_INVALID_NAF_ID	ERROR	
0xc0020041	RPC_NT_CANNOT_SUPPORT	ERROR	
0xc0020042	RPC_NT_NO_CONTEXT_AVAILABLE	ERROR	
0xc0020043	RPC_NT_INTERNAL_ERROR	ERROR	
0xc0020044	RPC_NT_ZERO_DIVIDE	ERROR	
0xc0020045	RPC_NT_ADDRESS_ERROR	ERROR	
0xc0020046	RPC_NT_FP_DIV_ZERO	ERROR	
0xc0020047	RPC_NT_FP_UNDERFLOW	ERROR	
0xc0020048	RPC_NT_FP_OVERFLOW	ERROR	
0xc0020049	RPC_NT_CALL_IN_PROGRESS	ERROR	
0xc002004a	RPC_NT_NO_MORE_BINDINGS	ERROR	
0xc002004b	RPC_NT_GROUP_MEMBER_NOT_FOUND	ERROR	
0xc002004c	EPT_NT_CANT_CREATE	ERROR	
0xc002004d	RPC_NT_INVALID_OBJECT	ERROR	
0xc002004f	RPC_NT_NO_INTERFACES	ERROR	
0xc0020050	RPC_NT_CALL_CANCELLED	ERROR	
0xc0020051	RPC_NT_BINDING_INCOMPLETE	ERROR	
0xc0020052	RPC_NT_COMM_FAILURE	ERROR	
0xc0020053	RPC_NT_UNSUPPORTED_AUTHN_LEVEL	ERROR	
0xc0020054	RPC_NT_NO_PRINC_NAME	ERROR	
0xc0020055	RPC_NT_NOT_RPC_ERROR	ERROR	
0xc0020057	RPC_NT_SEC_PKG_ERROR	ERROR	
0xc0020058	RPC_NT_NOT_CANCELLED	ERROR	

Co

Table 13.10 – continued from previous page

Code	Status	NTSTATUS severity	S
0xc0020062	RPC_NT_INVALID_ASYNC_HANDLE	ERROR	
0xc0020063	RPC_NT_INVALID_ASYNC_CALL	ERROR	
0xc0020064	RPC_NT_PROXY_ACCESS_DENIED	ERROR	
0xc0030001	RPC_NT_NO_MORE_ENTRIES	ERROR	
0xc0030002	RPC_NT_SS_CHAR_TRANS_OPEN_FAIL	ERROR	
0xc0030003	RPC_NT_SS_CHAR_TRANS_SHORT_FILE	ERROR	
0xc0030004	RPC_NT_SS_IN_NULL_CONTEXT	ERROR	
0xc0030005	RPC_NT_SS_CONTEXT_MISMATCH	ERROR	
0xc0030006	RPC_NT_SS_CONTEXT_DAMAGED	ERROR	
0xc0030007	RPC_NT_SS_HANDLES_MISMATCH	ERROR	
0xc0030008	RPC_NT_SS_CANNOT_GET_CALL_HANDLE	ERROR	
0xc0030009	RPC_NT_NULL_REF_POINTER	ERROR	
0xc003000a	RPC_NT_ENUM_VALUE_OUT_OF_RANGE	ERROR	
0xc003000b	RPC_NT_BYTE_COUNT_TOO_SMALL	ERROR	
0xc003000c	RPC_NT_BAD_STUB_DATA	ERROR	
0xc0030059	RPC_NT_INVALID_ES_ACTION	ERROR	
0xc003005a	RPC_NT_WRONG_ES_VERSION	ERROR	
0xc003005b	RPC_NT_WRONG_STUB_VERSION	ERROR	
0xc003005c	RPC_NT_INVALID_PIPE_OBJECT	ERROR	
0xc003005d	RPC_NT_INVALID_PIPE_OPERATION	ERROR	
0xc003005e	RPC_NT_WRONG_PIPE_VERSION	ERROR	
0xc003005f	RPC_NT_PIPE_CLOSED	ERROR	
0xc0030060	RPC_NT_PIPE_DISCIPLINE_ERROR	ERROR	
0xc0030061	RPC_NT_PIPE_EMPTY	ERROR	

A

Aggregation, 33, 121
Aggregation period, **127**
Alerting, 76
Application, 29, 70, 99, 105, **127**
Application NC, **127**
Application Port Range, **127**
Autopcap, 92

B

Bandwidth Chart, 13
BCA, 72, 81
BCN, 73, 82
Browser, 121
Business Critical Application, 72, 81
Business Critical Network, 73, 82
Byte, 27

C

Cacti, 131
Client, 31
Collector, **127**
Connection Time (CT), **127**
Conversation, 30, 121, **127**
CSV, 23

D

Dashboard, 81, 82, 86, 89
Data Transfer Time (DTT), **127**
Deduplication, 43, 122
Delta sessions, **127**
Destination, 31
Device Identifier, **127**
DiffServ Code Point (DSCP), **127**
Distributed Architecture, 45
DNS, 109, 122
DNS perfoarmance, 22
DTT, 97, 121

E

Email, 76
End User Response Time, **128**
EURT, 88, 96
Export, 23

F

Fallback, 27, **128**
Flow, **128**

G

Graphical Interface, 9

H

HTTP analysis, 14
HTTP hit, **128**
HTTP page, **128**
HTTPS, 79

I

ICMP, 23, 107
Initial Sequence Number, 122, **128**
IP merging, 30

J

Jitter, 85, **128**

K

KiB, 27

L

Language, 67
License, 111, 117
License Check, 118
Login, 11

M

Matrix, 16
Maximum Transfert Unit (MTU), **128**
Media Access Control (MAC) address, **128**
metric, 34
MiB, 27
Mirroring, 40, 43
MOS, 83, 85

N

Nagios, 134
Netflow, 50

O

Observation period, **128**
Open Source, 123
Operating System (OS), **128**

P

Packet Analysis, 92
Packet Loss, 85
PCAP, 92
PDF, 23, 74
Performance Chart, 12
Poller, **128**
Promiscuous mode, 49
Protocol, 41
Protocol Stack, **128**
Pulsar, 63, 66

R

Report, 74
Reset, 106
Restore, 64
Retransmission, 106, **128**
Retransmission Delay (RD), **128**
Retransmission Duplicate ACK, **128**
Retransmission Rate (RR), **128**
Retransmission Total, **129**
RFC
    RFC 1034, 109
    RFC 1035, 109
    RFC 3261, 83
    RFC 3435, 83
    RFC 3550, 83
    RFC 3551, 83
    RFC 3605, 83
Round Trip Time (RTT), **129**
RST, 106
RTCP, 83
RTP, 83
RTT, 96

S

Server, 31
Server Response Time (SRT), **129**
Session, 123, **129**
Shell, 63
SIP, 83
SNMP, 76
Source, 31
SRT, 97, 121
Subnet, **129**
Support, 66

T

TAP, 40
TCP, 122, 123
TCP events, 22
TCP Handshake, **129**
Tcpdump, 92, 122
Timeout, **129**
TLS, 79
Top reports, 21
Triggered PCAP, 94

U

Upgrade, 117
User, 67

V

VMWare, 47, 51, 123
Voice Quality, 83
VoIP, 83
VPN, 66

W

Web Application Pattern, **129**

Z

Zone, 27, 68, **129**